Quality Tactics

Develop Quality-Driven Solution Strategies for Software Architectures

Over 400 possibilities for achieving quality goals in software systems, their development and operation

Markus Harrer

Quality Tactics

Developing Quality-Driven Solution Strategies for Software Architectures

Markus Harrer

This book is available at https://leanpub.com/qualitytactics

This version was published on 2025-05-20

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Foreword

For many software architects, it is a great challenge to achieve an appropriate level of quality for a software system throughout its various lifecycle phases. The multitude of possibilities for introducing quality into a system can seem overwhelming and unstructured. Additionally, experiences with solutions may be lacking due to the specific usage scenarios of a system, often leading to suboptimal solutions adopted from other systems.

This book offers itself as a sparring partner for identifying suitable measures to enhance software quality. It concisely lists over 400 possibilities for achieving quality in software systems. The ISO 25010:2011, an international standard that defines quality models for evaluating the quality of software products and systems, serves as the basic structure for them. It encompasses various quality characteristics that help systematically describe diverse aspects of software quality. They support us in software architecture development by enabling discussions about specific quality requirements with various stakeholders and developing clearly defined quality goals.

These quality goals, along with the functional requirements and framework conditions, are essential influencing factors for creating the solution strategy of a software architecture. The solution strategy demonstrates how the software architecture tackles the challenges posed by the requirements using suitable principles, patterns, and other measures. These measures then significantly influence further working activities such as the development, maintenance, and operation of a software system.

The core topic of this book, comprehensively addressed under the term "quality tactics," is which measures potentially come into question for which quality goals. It supports software creators by presenting numerous possibilities for achieving quality goals. Many of them are already known but do not seem to be present in our minds at crucial moments. Therefore, the aim of the listing is to provide software architects with the widest possible range of ideas for their own individual solution strategy. The individual possibilities thus offer the starting point for developing the solution strategy and can have a valuable influence on individual architectural decisions.

In addition to quality tactics, this book also contains another approach to quality: quality illusions. These measures do not achieve quality goals in the original sense but rather give the impression that quality has been built into a system without the quality goals actually being achieved. They are placed in a separate chapter to distinguish them from the real quality tactics, as some illusions are quite debatable.

Have fun finding your own solution strategy!

Markus Harrer, July 2024

Acknowledgments

At this point, I would like to thank my current and former employers who, through a wide variety of situations, have always encouraged me to find suitable solutions for very individual quality challenges. As a result, I have come into contact with many of the measures listed here during my career, which has allowed me to structure them thematically within the individual chapters and evaluate them in terms of their effort, benefit, and popularity.

I would also like to express my gratitude to the numerous participants in my software architecture and software modernization training courses. The very individual questions and discussions here constantly motivate me to research, comprehend, and recommend appropriate answers to the sometimes very individual problems.

Also, a big thank you to my family, who make it possible for such a book to be created outside of working hours and who are always ready with advice and feedback on the content.

Use of AI

Without the support of a large language model, this book would not have been feasible for me in terms of time. To cope with the large number of possibilities and diversity, this work was therefore initially created in collaboration with Claude 3.5 Sonnet and Opus. The initially generated output, based on knowledge from the software architecture world, was curated, sorted, and improved by me to provide a reliable reference work for daily tasks in software architecture development. Additionally, for the English translation, ChatGPT 4o was used.

Albeit this might sound like an easy job, the opposite is the case: I put in days of work revising the texts and also wrote plenty of Python code to bring this book into the shape and consistency you can read now. I am now also constantly taking feedback for further editions and versions of the book in order to create a work that is extremely broad-based and up to date.

About this edition

This edition of the book provides an initial collection of quality tactics. I am convinced that it already offers significant value to software developers in implementing higher quality in our software systems. However, a book that gathers possibilities for improvement can never be complete. Therefore, future editions will likely follow to create an even more comprehensive list.

So if you have any tips for further quality tactics or feedback in general, please feel free to contact me at any time. The easiest way is via email to qualitytactics@markusharrer.de or via LinkedIn (https://www.linkedin.com/in/markus-harrer/).

Picture credits: The image on the cover (and in the print edition, also the image on the back cover) is from vectorjuice on Freepik and was slightly changed by me.

About Quality Tactics

Quality tactics help to achieve quality goals systematically and goal-driven. They allow software development to be structured and planned more effectively by specifying concrete measures for improving and ensuring quality. By applying such tactics, risks can be minimized, and the satisfaction of stakeholders (customers, end users, operators, product managers, and more) can be increased.

This book lists hundreds of quality tactics for various quality characteristics from the ISO/IEC 25010 quality standards. It serves as a source of inspiration for considerations on how to establish appropriate quality for a software system. Not only are technical possibilities listed, but also many possibilities surrounding the development of software systems that allow quality goals to be achieved.

In addition, the contents of the book can be used to evaluate the meaningfulness of existing ideas already present in software systems. Here, the result may also be that it would make sense to consider simpler alternatives to remove an already implemented quality tactic, as there are simpler means to achieve the quality goals.

The selection and combination of tactics depend on the specific requirements, framework conditions, and goals of the respective system and its objectives. And this is where the problem of many software development projects lies: it is often unclear what the goals are, especially in terms of quality! Because without goals, it is absolutely difficult to say whether you are on the right track or whether there is a quality deficit. This also makes it difficult to motivate the resulting costs for introducing and maintaining certain quality tactics. Therefore, before selecting quality tactics, I recommend getting a clear picture of the necessary qualities appropriate to the evolution stage of the software system. Here, I recommend quality requirements workshops such as Quality Storming or Mini-Quality Attribute Workshop.

In the following, you will find the quality tactics, including an indication of which quality characteristic of ISO/IEC 25010 the tactic particularly contributes to, a brief description, possible consequences, and synergy effects with other quality goals. This makes it possible to select those tactics that can have several positive effects on multiple quality goals to be achieved for a given set of different quality requirements.

Within the individual sections, the quality tactics are ordered, as far as meaningful, according to their descending familiarity (in the sense of "most frequently and most effectively used"), with the usual suspects or no-brainer solutions listed first. This allows readers to orient themselves based on the potentially already generally known and subsequently the not yet so widely known quality tactics in order to get a quick picture of the possibilities.

In addition, similar quality tactics have been grouped to give readers a quick picture of tactics of a certain type. If abstract quality tactics are described, they are detailed by subsequent, more concrete quality tactics. Due to the large number of quality tactics for the individual quality characteristics, there is no overlap-free list, as different tactics can simultaneously realize different quality goals. However, in these cases, the particular advantage of quality improvement with regard to a specific quality characteristic is highlighted for the respective quality tactic.

Enjoy finding the appropriate quality for the software system!

Functional Suitability

Functional suitability ensures that the software provides all necessary functions, that these functions work correctly, and that they are suitable for the users' requirements. It consists of various sub-characteristics:

- **(Functional) Completeness** ensures that no important functions are missing that are necessary to fulfill the users' needs. This should avoid situations where users have to find alternative solutions or workarounds because certain functions are not available, or they are disappointed because the software does not cover their functional requirements.
- **(Functional) Correctness** guarantees that functional errors and malfunctions are minimized. This increases users' confidence in the software, as they can rely on the required accuracy of the results.
- **(Functional) Appropriateness** ensures that the provided functions are actually useful for the users and correspond to their requirements. Unnecessary or superfluous functions that could make the software more complicated without providing additional benefits are avoided.

These qualities can be achieved through the following quality tactics, noting that many of the tactics listed here are outside the scope of software architects. This is in the nature of the quality characteristic "Functional Suitability".

Requirements Analysis

Systematic collection, analysis, and documentation of functional requirements

Requirements analysis forms the basis for the development of functionally suitable software. The needs and expectations of the stakeholders are captured, structured, and prioritized in detail. Systematic checks for completeness, consistency, and feasibility ensure that the requirements are correct and appropriate. The result is a specifications document approved by all parties involved, which serves as a binding foundation for further development.

Supports: Completeness, Correctness, Appropriateness
Consequences: High initial effort, but basis for focused development and high acceptance of the software.

`#RequirementsManagement` `#StakeholderAnalysis` `#RequirementsSpecification`

User Stories

Formulate requirements from the user's perspective

User stories clearly explain what a user needs to do in a certain situation to reach a goal. They follow the format "As a [role], I want [goal] so that [benefit]." This clear structure ensures the appropriateness of the requirements, as they directly address the needs of the users. This promotes communication between developers and end users and ensures that the developed solutions actually provide the desired value.

Supports: Appropriateness
Supports also: Operability
Consequences: Requires rethinking and practice in writing, risk of overly detailed stories.

`#UserStories` `#AgileDevelopment` `#UserFocus`

Prototyping

Gather early feedback on functionality and usability

Prototypes are simplified, incomplete versions of the software that demonstrate selected key features. They allow users and other stakeholders to experience, evaluate, and provide suggestions for improvement on the planned functionality at an early stage. Prototypes foster a shared understanding of the requirements and reduce the risk of misdevelopment.

Supports: Appropriateness
Supports also: Operability
Consequences: Additional effort for development and review of the prototypes.

`#Prototyping` `#Feedback` `#RiskMinimization`

Functional Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

User Acceptance Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Definition of Done

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Product Owner

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Personas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Feedback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Direct Feedback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain Experts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

On-Site Customer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Iterative Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feature Driven Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain-Driven Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Bounded Contexts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain-aligned Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business Process Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Standard Software

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Evolutionary Requirements Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Subject Matter Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Functional Debt Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Usability Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business Test Cases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain Quiz

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business Quality Scenarios

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Behavior-Driven Development (BDD)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Acceptance Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain-Specific Languages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Prototypes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Tracer Bullets

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Functional Spike

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Customizing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Decision Tables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business process automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Rule-Based Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Business Event Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Quality Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Value Range Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain-based Authorization Concept

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Domain Data Versioning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Microservices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Archiving

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Datensparsamkeit

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Enrichment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Ecosystems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Usability

Usability ensures that the software is easy to learn and use and provides a positive user experience. It consists of various sub-characteristics:

- **(Appropriateness) Recognizability** ensures that users can easily comprehend and understand the functions and operation of the software that they need. Clearly understandable instructions and help texts are essential here.
- **Learnability** describes how easy it is for new users to learn and efficiently use the software. This shortens the introduction time and increases user acceptance.
- **Operability** ensures that users can operate the software easily and efficiently. An intuitive user interface and logical navigation contribute to this, for instance.
- **(User) Error Protection** ensures that users are protected from errors when using the software and that errors can be easily corrected. This improves the user experience and reduces frustration.
- **(User Interface) Aesthetics** refers to the appealing and attractive design of the software. A well-designed user interface helps increase user satisfaction and intuitive operability.
- **Accessibility** guarantees that the software can also be used by people with disabilities. This includes accessibility standards and practices that ensure broad usability.

These quality tactics can provide valuable help for better usability:

User-Centered Design

Incorporate users' needs, expectations, and abilities from the beginning

User-centered design is an iterative process that focuses on the needs, expectations, and abilities of users. Through methods such as interviews, observations, and usability testing, user requirements are captured and integrated into the development process. The goal is to develop software that is optimally tailored to users and exhibits high usability. By involving users early and continuously in the development process, problems can be identified and resolved at an early stage.

Supports: Learnability, Operability, Aesthetics
Consequences: Higher effort in requirements analysis and usability testing, but long-term more satisfied users and less rework.

`#UserCenteredDesign` `#UsabilityEngineering` `#UserRequirements` `#Iterative`

Usability Tests

Conducting tests with representative users

In usability tests, the software is tested by representative users under realistic conditions. Problems with operation, ambiguities in navigation, or missing features are identified. The insights gained are incorporated into the further development of the software to continuously improve usability. Usability tests can be conducted at various stages of the development process, both with prototypes and with finished software versions.

Supports: Operability, Learnability, Recognizability
Consequences: Additional effort for conducting the tests and evaluating the results, but early detection and resolution of usability issues.

`#UsabilityTesting #UserFeedback #Prototyping #IterativeImprovement`

Wireframing

Create preliminary visual representations as a basis for discussion

Wireframing represents UI controls and functional flows through visual sketches. This method allows for the early evaluation of design and user navigation, helping to identify potential usability issues. Involving stakeholders (especially developers and users) in the wireframing process ensures that all relevant aspects of usability are considered. The results of the wireframing process provide valuable insights into user expectations and areas needing improvement, supporting the development of user-friendly solutions.

Supports: Operability
Consequences: Additional effort for the creation and revision of sketches.

`#Wireframing #UserFeedback #DeveloperFeedback`

Responsive Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Mobile First Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Consistent User Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Style Guide

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Consistent Terminology

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cognitive Load Minimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Intuitive Navigation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Icons

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Visual Hierarchy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Progressive Disclosure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Plain Language

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Understandable Error Messages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Real-time Input Validation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Contextual Help

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feedback

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Gamification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Customizable User Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Custom Views

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Adaptive Behavior

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Localization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Performance Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Asynchronous Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Single Page Application

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Progressive Web App

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Keyboard Support

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Search Function

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Integrated Onboarding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Interactive Tutorials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Video Tutorials

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Frequently Asked Questions (FAQ)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feedback Mechanisms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

A/B Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

User Communities

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Personal Support

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Knowledge Base

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Accessibility Concept

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Assistive Technology Support

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Adjustable Font Sizes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

High Color Contrasts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Subtitles and Transcripts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Reliability

Reliability ensures that the software functions without errors under defined conditions over a certain period of time. It consists of various sub-characteristics:

- **Maturity** describes the ability of the software to deliver stable performance and minimize errors in the application. This increases users' confidence that the software will function consistently.
- **Availability** ensures that the software is accessible and usable whenever it is needed. This minimizes downtime and maximizes the application's uptime.
- **Fault Tolerance** ensures that the software is able to continue functioning in the event of errors or unexpected events. This increases the software's resilience against failures.
- **Recoverability** guarantees that the software can quickly return to an operational state after an error or failure. This reduces downtime and the impact of disruptions on users.

These quality tactics can be considered for more reliable software systems:

Error Handling

Mechanisms for detecting, logging, and handling errors

Through robust error handling, exceptions and unexpected situations can be caught without causing the entire system to crash. Errors are logged to analyze root causes and prevent future issues. Appropriate error messages also inform users about current problems. Structured error handling enables the system to recover from error states. Error handling increases the reliability and availability of the system and minimizes the impact of errors on users.

Supports: Fault Tolerance
Consequences: Increased implementation effort and complexity due to additional error handling code.

`#ExceptionHandling` `#Logging` `#ErrorManagement` `#Resilience`

Redundancy

Multiple instances of critical components or systems

Redundancy ensures that critical components or entire systems are duplicated, allowing another component to seamlessly take over in case of failure. This avoids single points of failure and increases overall system availability. Careful planning and monitoring of redundant components are necessary to achieve the desired reliability.

Supports: Availability
Consequences: Increased costs and complexity due to additional hardware, software, and maintenance.

`#FaultTolerance` `#Failover` `#HighAvailability` `#Spare`

Exceptions

Using exceptions for signaling and handling error states

Exceptions signal and handle error states in a system. When a component encounters a situation it cannot process normally, such as invalid input, resource shortages, or internal errors, it can trigger an exception. This interrupts the normal program flow and transfers control to a special exception handling routine. There, the error can be logged, alternative actions taken, or the exception passed to a higher level. Exceptions enable a clear separation of normal code and error handling, as well as structured error state management.

Supports: Fault Tolerance
Consequences: Increased complexity due to the implementation of exception handling routines and potential performance degradation with frequent exceptions.

`#ExceptionHandling` `#ErrorHandling` `#Robustness` `#DefensiveProgramming`

Checklists

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Runbooks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Error Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Monitoring System Utilization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Boring Technologies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Resilience

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Disaster Recovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

On-Call Duty

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Load Balancing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Chaos Engineering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Elastic Resource Utilization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Proactive Capacity Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Service Level Agreements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Service Level Objectives

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Service Level Indicators

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Replication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Failover Mechanisms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Self-Monitoring and Diagnosis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Isolation of Faulty Components

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Environment Parity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Production environment maintenance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Site Reliability Engineering (SRE)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Graceful Degradation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Circuit Breaker

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Bulkhead

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Self-Test

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Ping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Heartbeat

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Transactions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Retry

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Watchdog

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Smoke Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Nonstop Forwarding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Timestamping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Status Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Failover Cluster

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Redundant Data Storage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Rollback Mechanisms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Blue-Green Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feature Toggles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Rolling Updates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dark Launches

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Canary Releases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fault-Tolerant Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fault Containment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Error Correction Codes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Error Reporting and Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Checksums

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Redundant Checksums

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Data Verification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Plausibility Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Restore Points

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Monitoring System Integrity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Load Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Regular Backups

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Incident Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Root Cause Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

High Availability Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Error Logs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Integrity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Software

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Regular Maintenance and Updates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Integration and Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Immutable Infrastructure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Automated Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Performance Efficiency

Performance Efficiency ensures that the software optimally uses the available resources and performs well under load conditions. It consists of various subcharacteristics:

- **Time-behavior** describes how quickly the software responds to inputs and completes tasks. This includes loading times and response times, which are crucial for a positive user experience.
- **Resource Utilization** ensures that the software efficiently uses the available system resources, such as CPU, memory, and network bandwidth. This reduces resource consumption and increases performance.
- **Capacity** guarantees that the software can handle a certain amount of data or users without losing performance. This is important for the scalability and future-proofing of the application.

The following quality tactics can fulfill requirements regarding performance efficiency:

Caching

Caching frequently needed data

Through caching, data that is often read but rarely changed is kept in a fast cache after the first access. This can be done in memory, but also on fast SSDs or dedicated cache servers. On subsequent accesses, the data can be read directly from the cache without burdening slower hard drives or databases. Caching reduces response times and increases system throughput.

Supports: Time-behavior, Resource Utilization
Consequences: Increased memory usage, risk of outdated data with insufficient cache invalidation.

`#Caching` `#PerformanceOptimization`

Load Balancing

Distribution of the load across multiple parallel processing units

Load balancing distributes requests and tasks evenly across multiple servers, processors, or threads. This enables optimal utilization of available resources and helps to avoid bottlenecks. By using dedicated components such as load balancers or through implementation within the application itself, performance efficiency is increased. In particular, the response time is improved, as processing speed is enhanced while consumption behavior is optimized through even resource utilization.

Supports: Time-behavior, Resource Utilization
Supports also: Capacity
Consequences: Increased complexity, necessity for synchronization and state management.

`#LoadDistribution #LoadBalancing #Scalability`

Asynchronous Processing

Decoupling of calls and execution through asynchronicity

Asynchronous processing does not handle requests and tasks directly; instead, it first queues them. This allows the calling component to be released immediately, enabling it to make further requests. The processing is carried out by separate processing units, which increases concurrency and avoids blocking calls. This technique optimizes performance efficiency by reducing response times and minimizing resource consumption, leading to an overall improvement in system performance.

Supports: Time-behavior, Resource Utilization
Supports also: Capacity
Consequences: Increased complexity, necessity for error handling and monitoring of processing.

`#AsynchronousProcessing #Decoupling #Concurrency`

Asynchronous Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Performance Budgets

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Resource usage optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compression

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

API calls optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Efficient algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Database Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

NoSQL Databases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Graph Databases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Load Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Stress Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Profiling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Resource Pooling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Parallelization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Distributed Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Concurrency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pipelining

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Streaming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Elastic Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Vertical Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Horizontal Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Content Delivery Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Static Site Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Server Side Rendering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Edge Computing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Client Side Rendering

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Batch Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

In-Memory Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Lazy Loading

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Deferred Static Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Lazy Evaluation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Adaptive Streaming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Progressive Loading

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pagination

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Infinite Scrolling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Virtualized Lists

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Approximation methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Neural Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Predictive Loading

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Predictive Prefetching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code splitting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Tree Shaking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Specialized hardware

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Performance Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Performance Measurements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Static Code Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dynamic Programming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Probabilistic Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Vectorization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Genetic Algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Quantum Computing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Quantum-Optimized Algorithms

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Performance Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Transparent Performance Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Serverless Computing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Reactive Programming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Mass Test Data Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Memory Hierarchy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Distributed Caching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Partitioning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Deduplication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Aggregation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Materialized Views

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Sampling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Archiving

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Stream Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security

Security ensures that the software protects the confidentiality, integrity, and availability of data and is secured against unauthorized access and attacks. It consists of various subcharacteristics:

- **Confidentiality** ensures that sensitive data can only be accessed and processed by authorized users. This protects against data loss and theft.
- **Integrity** guarantees that data cannot be altered without authorization during storage and transmission. This ensures the accuracy and reliability of the data.
- **Non-repudiation** ensures that all security-relevant actions in the software can be traced and verified. This includes logging access and changes.
- **Accountability** ensures that all actions can be uniquely attributed to a user or entity. This increases responsibility and enables the tracing of incidents.
- **Authenticity** ensures that the identity of users, systems, and data is verified and confirmed. This protects against identity theft and ensures that communication and transactions occur between trusted parties.

The following measures can make software systems more secure:

Encryption

Encrypt data during transmission and storage

Encryption is an essential process for securing confidential information against unauthorized access. It transforms readable data into unreadable formats using algorithms so that only authorized users with the correct key can decrypt the data. Proven algorithms such as AES or RSA are employed, which provide additional security through sufficiently long keys. Encryption is used both for the transmission of data over networks and for storage in databases or files.

Supports: Confidentiality
Consequences: Performance degradation, increased complexity, effort for key management.

`#Cryptography` `#DataSecurity` `#KeyManagement`

Authentication

Verify the identity of users and systems

Authentication is the process by which users and systems prove their identity to gain access to protected resources. Common mechanisms include usernames and passwords, which are the most frequently used, as well as two-factor authentication, which provides an additional layer of security by requiring a second proof, such as a code on a mobile device. Other methods include digital certificates and biometric features like fingerprints or facial recognition.

Supports: Authenticity
Consequences: Additional effort for users, complexity of authentication procedures.

`#IdentityManagement #AccessControl #AuthenticationProtocols`

Two-Factor Authentication

Verify identity using two independent factors

Two-factor authentication (2FA) is a security procedure that goes beyond the conventional password or PIN entry. In addition to the knowledge provided by a password, a second factor is required to grant access. This second factor can be a TAN, a token, or a biometric characteristic such as a fingerprint. Both factors must be entered correctly to allow access to an account or system. This additional layer of security significantly reduces the risk of unauthorized access.

Supports: Authenticity, Integrity
Consequences: Additional effort for users and administration, costs for hardware tokens.

`#2FA #Multifactor #Authentication`

Authorization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Authorization Concept

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Role-Based Access Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Least Privilege

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Logging and Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Input Validation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Session Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Community

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Policies for Users

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Policies for Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Training

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Raising User Awareness

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Network Segmentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Software Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security by Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Certification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Audits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Risk Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Incident Response Measures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Backup and Recovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Protocols

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Datensparsamkeit

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Threat Intelligence

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Red Teaming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Architecture Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Digital Forensics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Honeypots

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Threat Modeling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Abuse Case Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Requirements Definition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure by Default

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Trust Boundaries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Flow Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cryptographic Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Key Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Coding Guidelines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Static Code Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dynamic Code Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Secure Programming Interfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Prepared Statements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Output Encoding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Canonicalization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fuzz-Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Negative Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Regression Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Tests by External Parties

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Penetration Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

System Hardening

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Patch Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Defense Lines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Malware Protection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Monitoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Endpoint Detection and Response

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Vulnerability Scans

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Third-Party Dependency Check

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Configuration Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Emergency Drills

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security-Relevant Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Incident Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Physical Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Security Culture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Maintainability

Maintainability ensures that the software is easy to modify, extend, and maintain. The following more specific quality characteristics can be distinguished here:

- **Modularity** refers to the structuring of the software into independent modules that can be developed, tested, and maintained separately. This simplifies the management of the software and allows for easier and less error-prone changes.
- **Reusability** guarantees that components of the software can be reused in different applications or contexts. This reduces development effort and increases efficiency by utilizing proven components.
- **Analyzability** describes the ability to analyze and understand the software and its source code. This facilitates troubleshooting and changes.
- **Modifiability** ensures that changes to the software can be made easily and without undesirable side effects. This increases the flexibility and adaptability of the software.
- **Testability** ensures that the software is easy to test or generally verify, enabling early detection and correction of both functional and architectural errors. This improves the overall quality of the application.

The following quality tactics can be considered here:

Domain-Driven Design

Structuring software architecture based on the business domain

Domain-Driven Design (DDD) is an approach to software development that places the business domain at the center of the design process. Developers work closely with subject matter experts to identify the relevant concepts and processes and integrate them into the system. By creating a shared model, the software structure becomes clearer and more understandable. DDD promotes modularity and allows for flexible adaptation to changing requirements, which improves the long-term maintainability of the software.

Supports: Modularity
Supports also: Analyzability, Appropriateness
Consequences: High initial effort for domain analysis, potential over-modeling in simple applications.

`#UbiquitousLanguage #BoundedContexts #AggregateDesign`

Separation of Concerns

Divide functionalities into clearly defined, independent areas

The separation of concerns is a fundamental principle in software architecture that aims to divide an application into distinct modules or components. Each module takes on a specific task or responsibility, thereby reducing the complexity of the application. This clear separation promotes clarity, facilitates understanding of the code, and allows developers to make changes to individual functionalities without affecting the entire system.

Supports: Modularity
Supports also: Modifiability, Testability
Consequences: Initial additional effort in structuring, potential over-architecture in small applications.

`#ModularDesign` `#SingleResponsibilityPrinciple` `#ComponentBasedDevelopment`

Modularization

Divide application into small, independent, and reusable components

Modularization divides an application into separate, loosely coupled modules with clearly defined interfaces. Each module encapsulates a specific functionality and has minimal dependencies on other modules. This approach reduces complexity, as individual modules are easier to understand, maintain, and test. Changes to one module have less impact on other parts of the system. The independent development and reuse of modules increase the efficiency and flexibility of the development process.

Supports: Modularity
Supports also: Reusability, Testability
Consequences: Increased initial planning effort, potential performance drawbacks due to interface communication, necessity for consistent module management.

`#ComponentBasedDevelopment` `#LooseCoupling` `#InterfaceDesign`

Bubble Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Modulith

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Layered Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pattern Language

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Anti Corruption Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Walking Skeleton

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Technical Spike

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Clean Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

SOLID Principles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Refactoring

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Refactoring Katas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Static Code Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Strategic Code Deletion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Deprecation Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Collaborative Problem Solving

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pair Programming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fair Source

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Conventions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Quality Gates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Automated Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Integration Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Test-Driven Development (TDD)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Behavior-Driven Development (BDD)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Coverage Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dependency Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dependency Injection Container

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dependency Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Agile Development Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Workshops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Review Board

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Governance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Conformity Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fitness Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Roadmap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Version Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feature Toggles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Knowledge Management System

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Architecture Decision Records (ADR)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Living Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Docs as Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

API-First Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

API Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Comments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fluent Interfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Mutation Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Property-Based Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Aspect-Oriented Programming (AOP)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Code Generation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Delivery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Microservices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Containerization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Infrastructure as Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility

Compatibility ensures that the software works in different environments and with other systems and applications. It consists of various subcharacteristics:

- **Co-existence** describes the ability of the software to function alongside other independent software products on the same system without conflicts. This increases flexibility and usability in different environments.
- **Interoperability** ensures that the software can communicate and exchange data with other systems and applications. This enables seamless integration into existing infrastructures and improves overall functionality.

These qualities can be achieved through the following quality tactics:

Standardized Interfaces

Implement interfaces according to widely accepted standards

Using well-known communication protocols like REST or SOAP enables easier integration with other systems and technologies, as these standards are widely adopted and well-documented. By employing standardized interfaces, interoperability is increased, and the effort required to connect to other software components is reduced, as a variety of standard libraries are also available.

Supports: Interoperability
Consequences: Limitation of flexibility in interface design, overhead due to an additional abstraction layer.

`#Interfaces` `#Interoperability` `#Integration`

Protocol Abstraction

Decoupling communication protocols through abstraction

Instead of directly accessing concrete protocols like HTTP or TCP/IP, abstract layers are introduced that are independent of the underlying protocols. This allows communication to be flexibly adapted to different protocols without needing to change the program code. Protocol abstraction increases compatibility and facilitates switching between different communication protocols.

Supports: Co-existence
Consequences: Increased complexity due to additional abstraction layers, potential performance loss.

`#Protocols` `#Abstraction` `#Decoupling`

Data Formats

Use standardized and widely adopted data formats for data exchange

Data formats play a crucial role in the exchange of data between different systems. Formats such as XML, JSON, or CSV are widely used and supported by numerous applications. Proprietary or exotic formats significantly limit interoperability. By choosing common data formats, the exchange of information between different systems is simplified, which increases compatibility and promotes integration.

Supports: Interoperability
Consequences: Limitation of flexibility in data modeling, potential overhead due to conversions.

`#DataFormats` `#DataExchange` `#Interoperability`

Versioning Scheme

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Version Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Loose Coupling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Configurability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Abstraction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Adapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Facades

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Anti Corruption Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Bridges

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Mediator

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Microservices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Containerization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Virtualization and Containerization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Emulation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Standardized Protocols

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Serialization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Layers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Data Format Conversion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Version Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

API Versioning Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Backward Compatible APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Backward Compatibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Forward Compatibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Feature Flags

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Isolated Test Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Simulation Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Documentation of Compatibility Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Matrix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Guidelines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Testing by Users

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Metrics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Audits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Backward-Compatible Schema Migrations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Browser Compatibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Backlog

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Roadmap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Champions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Interoperability Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Consumer Driven Contracts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Error Message

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Backward Compatible Data Formats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Risk Assessment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Guidelines for Third-Party Providers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Certification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Smoke Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Testing Before Releases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Compatibility Criteria in Definition of Done

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Continuous Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Portability

Portability ensures that the software can be easily transferred from one environment to another. It consists of various subcharacteristics:

- **Adaptability** describes the ability of the software to function in different environments without requiring extensive changes. This includes various operating systems, hardware platforms, and configurations.
- **Installability** ensures that the software can be installed easily and without issues. A smooth installation process increases user satisfaction and reduces the need for IT support.
- **Replaceability** guarantees that the software can be easily replaced by another software with similar functionality. This increases flexibility and allows for the transition to better or more cost-effective solutions.

These quality tactics help to provide software systems with better portability:

Containerization

Packaging applications and their dependencies into isolated containers

Containerization allows software applications to be packaged together with all necessary libraries, configuration files, and dependencies into standardized units. These containers can then be executed on any system that supports the container runtime environment, regardless of the underlying operating system or hardware. This significantly increases portability, as the application runs consistently in different environments without requiring adjustments. Additionally, it simplifies deployment and scaling processes.

Supports: Adaptability
Supports also: Resource Utilization, Modifiability
Consequences: Increased initial effort for container creation and management, potential performance overhead due to additional abstraction layer.

`#Docker` `#Kubernetes` `#Microservices`

Platform-Independent Programming Languages

Using programming languages that run on different systems without modifications

Platform-independent programming languages like Java or Python enable the development of applications that can be executed on various operating systems and hardware architectures. These languages use virtual machines or interpreters that act as an abstraction layer between the program code and the underlying system. This allows the same source code to be compiled and executed on different platforms without requiring extensive modifications.

Supports: Adaptability
Supports also: Reusability
Consequences: Possible performance losses compared to native code, dependency on the availability and updating of the runtime environment.

`#Java #Python #CrossPlatform`

Externalized configuration

Separate environment-specific settings and application logic

By using external configuration files, environment-specific settings such as database connections, server addresses, or API keys are separated from the actual application logic. This allows the same application to be deployed in different environments (development, testing, production) by simply adjusting the configuration files. Modern frameworks and tools support this approach and facilitate the management of various configurations for different environments.

Supports: Adaptability
Supports also: Modifiability, Confidentiality
Consequences: Increased administrative effort for configuration files, necessity for secure storage of sensitive configuration data.

`#ConfigurationManagement #EnvironmentControl #Flexibility`

Portability Checklists

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Standardized Data Formats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Microservices Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cloud-native Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Abstraction Layers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Virtualization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

API-First Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Dependency Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Build Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform UI Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Database Abstraction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Object-Relational Mapping (ORM)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Configuration Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Portable Binary Formats

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Environment Variables for Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform Independence

Make software executable on different systems and environments without modifications

Platform independence increases the portability of software through the use of platform-independent programming languages, frameworks, and libraries. Platform-specific code is avoided or encapsulated, which minimizes adjustments. This architecture promotes coexistence in heterogeneous environments, as the software can be seamlessly operated on different platforms. This maximizes transferability and enhances flexibility in usage.

Supports: Installability
Supports also: Adaptability
Consequences: Higher development effort, possible limitations in the use of platform-specific functions.

`#PlatformIndependence` `#Portability` `#Adaptability`

Platform-Independent Scripting Languages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Virtual Development Environments

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Data Storage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Standardized Deployment Scripts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Build Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Abstracted File System Access

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Logging Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Configuration Files

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Virtual Networks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Test Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Containerized Databases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Encryption Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Platform-Independent Time Zone Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Graphics Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Build Scripts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Cross-Platform Package Managers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Quality Illusions

This section is very specific, and I debated for a long time whether to include it in the book. It addresses quality illusions: measures aimed at creating a better perception of quality through clever techniques than what is actually present. In some cases, these types of solution strategies are useful (e.g., when finding the right market fit for a solution or dealing with undemanding users).

Quality illusions are not, per se, quality tactics, as they merely simulate quality. They could negatively be viewed as hacks, dirty tricks, and workarounds. However, I find that too derogatory, as they can sometimes provide exactly the right "quality experience" required. Some approaches might be seen as morally questionable or even as fraud/intentional deception in certain contexts. Thus, it depends on the company's risk appetite whether to employ these kinds of quality illusions or not! Therefore, I explicitly do not recommend the use of specific quality illusions. Their use is at one's own risk (which generally applies to quality tactics as well, as I cannot make recommendations without the specific context).

Hence, quality illusions must be coordinated with both product management and customers. It must be a decision driven by product development to seemingly meet quality requirements. If software architects do this alone, they could find themselves in trouble and might even lose their jobs, as they could be labeled as shoddy workers. There are several risks associated with the use of quality illusions:

1. **Loss of Trust**: If users or customers see through the deception, it can lead to a significant loss of trust. Trust in a product or company is often hard to regain once lost.
2. **Legal Consequences**: Depending on the nature and extent of the deception, legal issues may arise, particularly concerning data protection or consumer deception.
3. **Quality Risk**: There is a danger that the actual project goals or quality standards are not met because the focus is on deception rather than genuine improvement.
4. **Cost Risk**: Long-term costs may increase if problems arise due to a lack of real quality measures and need to be addressed.
5. **Schedule Risk**: If feigned measures do not solve the actual problems, it can lead to delays in the project timeline.
6. **Ethical Concerns**: The use of deceptive tactics raises ethical questions and can damage a company's image.
7. **Ineffective Risk Management**: By focusing on feigned measures, genuine risks may be overlooked or inadequately addressed.
8. **Customer Satisfaction**: In the long run, user and customer satisfaction may suffer if the promised quality is not actually delivered.
9. **Competitive Disadvantage**: Companies that focus on real quality improvements can gain a competitive advantage in the long term over the ones that cheat.

10. **Internal Conflicts**: Employees who are aware of the deception may become demotivated or express ethical concerns (and might even leave the company because of it!).

Therefore, my advice is to rely on genuine quality tactics and play with honest cards!

Skeleton Screens

Displaying placeholder layouts during loading

Skeleton screens display a simplified version of the content layout before the actual data is loaded. This visual preview of the expected structure of the page creates the impression of faster loading times and reduces the perceived waiting time. Through engaging design, an aesthetic is created that captivates the user during the transition phase and conveys a sense of progress, even when the content is not yet available. In this way, illusions of quality are generated that enhance the user experience.

Supports: Aesthetics (seemingly)
Supports also: Time-behavior (seemingly)
Consequences: Increased development effort for creating the skeleton layouts.

`#SkeletonUI` `#LoadingOptimization` `#PerceivedPerformance`

Optimistic UI Updates

Immediate display of user actions before server confirmation

Optimistic UI updates create a seamless user experience by immediately reflecting user actions in the interface before confirmation from the server arrives. This approach conveys the impression of an extremely fast response time and enhances usability. The application assumes that the action is successful and updates the UI accordingly. In the event of a failure, the UI is reset, and the user is informed, keeping the interaction smooth and engaging.

Supports: Operability (seemingly)
Supports also: Time-behavior (seemingly)
Consequences: Risk of inconsistencies between UI and actual state, necessity of rollback mechanisms.

`#OptimisticUI` `#FastResponse` `#UserInteraction`

Shimmer Effect

Animated placeholders for content not yet loaded

The shimmer effect marks areas with loading content through shimmering animations instead of static loading bars or spinners. This subtle movement conveys a sense of activity and progress, significantly reducing the perceived wait time. Users remain engaged while the actual data is loaded in the background. This visual illusion creates a perception of quality that optimizes time-behavior and enhances the user experience.

Supports: Time-behavior (seemingly)
Supports also: Aesthetics (seemingly)
Consequences: Increased development effort for creating and animating the shimmer effects.

`#ShimmerUI` `#LoadingAnimation` `#VisualFeedback`

Micro Interactions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fake Progress Bar

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Navigation Maze

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pseudo-Personalization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Phantom Notifications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Simulated Real-Time Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Artificial Delays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pseudo-AI Interactions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Phantom Functionality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Artificial Learning Curve

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fake Localization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Pseudo-Multitasking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Fake Datensparsamkeit

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/qualitytactics.

Concluding Remarks

This concludes the over 400 possibilities that enable software architects and developers to develop targeted solution strategies based on quality goals. Depending on the tactic, it is now appropriate to delve deeper into each topic to approach the correct implementation, including assessing possible consequences. The internet and libraries are full of additional information.

At this point, I would like to once again encourage you to deeply consider whether a quality tactic can truly address the given quality requirements appropriately. Otherwise, absolutely over-engineered systems are created, which are no fun to continue developing.

Therefore: Find the right balance of quality in your software systems!

Let's go!

Markus Harrer