Principles of Practice
Development strategy

Robert Dupuis & Normand Séguin - LRGL, UQAM

Alain Abran & Kenza Meridji - GELOG, ETS
Objectives of the Initiative

• Fill gaps identified in the list of principles
• Strengthen the definitions and descriptions of principles
• Build consensus around the list and the process
• Validate results with the Computer Society
Description of Each Principle

• Each principle could include:
  – A “bumper sticker” phrase
  – An explanation of the principle
  – An explanation of how it is practical in different contexts
  – An explanation of the outcome of the principle
  – An explanation of why the outcome is good
  – Instances where the principle might not be appropriate
  – Consequences of applying and not applying the principle
  – (Perhaps this should be similar to the descriptions of “design patterns”)
  – Should be widely adopted and accepted

• For eventual Computer Society use, each principle should be cross-referenced to:
  – The process reference model
  – Knowledge in the SWEBOK Guide
  – Tasks in the CSDP Test Specification
  – IEEE and ISO/IEC standards that support implementation
Strategy Element: Principles of Practice

- Needs
  - Differentiating principles ("what to do") and standards ("how to do it")
  - Differentiating principles ("prescriptive") and knowledge ("descriptive")
  - Mechanism for consensus
  - Mechanism for maintenance and obsolescence
  - Criteria for principles
  - Characterization of the scope – (same as SWEBOK?)
  - Characterization of desired level of abstraction
  - Estimate of target number of principles
Consensus Building Models

• Use alternating Delphi/workshop model to formulate a set of principles and their summary-level explanation?
  – It’s better for “allocation” problems
  – It provides the opportunity for “celebrity endorsement”
• Use expert-mediated commenting [like SWEBOK Guide] to complete the explanations?
• End with an approval ballot?
• Use a Wiki approach?
Wiki

- (wi:.ki: <wee-kee>) or (‘w•.ki <wick-ey>)
- “A type of Web site that allows users to easily add and edit content and is especially suited for collaborative writing.” (Wikipedia)
- A tool for creating and controlling the evolution of an open source document
Proposed structure

• Baselined principle (not accessible for edition on line)

• Evolving version
  – Accessible for modification by participants
  – Some control on the soundness of proposed modifications

• Validation mechanisms
  – First level: according to criteria, by FP team
  – Second level: by group of participants
Status on the principles

• From the literature survey, 34 candidate principles (See Séguin 2006)
• Most of these 34 candidate principles have been provided with:
  – Description/explanation
  – How to apply
  – Consequences if not applied
• Consistent with definitions and criteria for identifying FP
Example

• Grow systems incrementally (Davis)
  – Explanation:
    • One of the most effective techniques to reduce risk in building software is to grow it incrementally.
  – Consequences:
    • The disadvantage is that, if an inappropriate architecture is selected early, a complete redesign may be necessary to accommodate later changes. Reduce this risk by building throwaway prototypes prior to starting the incremental development.
  – How to apply:
    • Start small, with a working system that implements only a few functions. Then grow it to cover larger and larger subsets of the eventual functionality.
Status of the Initiative on the Principles

• Identification of engineering criteria (see Meridji 2007)
• Identification of 9 software engineering principles
• Identification of the hierarchy of the set the other 25 candidates.
• Coverage analysis of the 9 principles in the SWEBOK Guide.
Next steps

• Experts workshops
• Set up the wiki site
  – Invite group (to be determined) to propose principles, or sources, for areas not covered by current list
  – Have larger community comment on the list though the wiki
• Link to Computer Society’s needs