

READY

WHY MOST SOFTWARE PROJECTS
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

हिंदी संस्करण

इस पुस्तक को ट्वीट करें!

कृपया [Twitter](#) पर इस पुस्तक के बारे में बताकर Luniel de Beer और Max Guernsey, III की मदद करें!

इस पुस्तक के लिए सुझाया गया ट्वीट है:

मैंने अभी Ready खरीदी — एक ऐसी किताब जो सॉफ्टवेयर लीडर्स और टीमों के लिए है जो सॉफ्टवेयर डिलीवरी में पुनः कार्य, अग्रेषण और असंरेखण को समाप्त करना चाहते हैं। [#CodeReady](#)

इस पुस्तक के लिए सुझाया गया हैशटैग [#CodeReady](#) है।

इस हैशटैग को Twitter पर खोजने के लिए इस लिंक पर क्लिक करके जानें कि अन्य लोग पुस्तक के बारे में क्या कह रहे हैं:

[#CodeReady](#)

तैयार (हिंदी संस्करण)

अधिकांश सॉफ्टवेयर प्रोजेक्ट्स क्यों विफल होते हैं और इसे कैसे सुधारें

Luniel de Beer और Max Guernsey, III

यह पुस्तक <https://leanpub.com/ready-hi> पर उपलब्ध है

यह संस्करण 2025-10-22 को प्रकाशित किया गया



यह एक **Leanpub** पुस्तक है। Leanpub लेखकों और प्रकाशकों को लीन पब्लिशिंग प्रक्रिया से सशक्त बनाता है। **लीन पब्लिशिंग** प्रगति में चल रही ई-बुक को हल्के टूल्स और कई संशोधनों का उपयोग करके प्रकाशित करने की प्रक्रिया है, जिससे पाठक की प्रतिक्रिया मिलती है, सही पुस्तक मिलने तक बदलाव किया जाता है और एक बार सही होने पर आकर्षण बनाया जाता है।

© 2025 Luniel de Beer और Max Guernsey, III

योहान वान आर्ट की याद में, जिन्होंने मेरे जुनून को पहचाना, मुझे वास्तविक प्रोग्रामिंग से परिचित कराया, और मुझे एक नए घर का रास्ता दिखाया। और मेरे माता-पिता के लिए, जिनका अटल समर्थन—मेरे पहले ग्राहक से लेकर अमेरिका में मेरे पहले घर तक—ने यह सब संभव बनाया।

—लुनिएल

मेरे परिवार के लिए, जिनके साथ सूरज उगता और डूबता है।

—मैक्स

विषय सूची

इस पुस्तक के बारे में	i
यह किसके लिए है	ii
इस पुस्तक का उपयोग कैसे करें	iii
लेखकों के बारे में	iv
प्रस्तावना	v

भाग I: कुछ तो कम है 1

अध्याय 1: छिपी हुई समस्या	2
अध्याय 2: आधारशिला की कमी का मूल्य	15
अध्याय 3: आवश्यकता परिपक्वता प्रवाह (Requirements Maturation Flow - RMF) का परिचय	28
अध्याय 4: क्या यह एजाइल है?	30

भाग II: तैयारी के लिए स्थान बनाना 31

अध्याय 5: पहला विस्तार	32
अध्याय 6: लोग ऐसा क्यों नहीं करते?	33
अध्याय 7: स्पष्ट तैयारी कार्य (RMF 1)	35
अध्याय 8: RMF 1 के प्रभाव	37

अध्याय 9: आरएमएफ 1 को व्यवहार में लाना	38
--	----

भाग III: कार्य पूर्णता का नियंत्रण 41

अध्याय 10: अगली आवश्यकता	42
------------------------------------	----

अध्याय 11: लोग आमतौर पर क्या करते हैं	44
---	----

अध्याय 12: कार्य-समाप्ति की परिभाषा को परिभाषित करना	46
--	----

अध्याय 13: विशिष्ट कार्य-समाप्ति परिभाषा (RMF 2)	49
--	----

अध्याय 14: RMF 1 और 2 के साथ जीवन	53
---	----

अध्याय 15: RMF 2 का संस्थापन	55
--	----

भाग IV: गेटिंग कार्यान्वयन 57

अध्याय 16: अंतिम आवश्यकता	58
-------------------------------------	----

अध्याय 17: रेडीनेस की परिभाषा की पृष्ठभूमि	60
--	----

अध्याय 18: कार्य-तत्परता परिभाषा को परिभाषित करना	62
---	----

अध्याय 19: तत्परता की विशिष्ट परिभाषा (RMF 3)	64
---	----

भाग V: संश्लेषण 68

अध्याय 20: <u>ज्यादातर</u> समय सीमाएं मायने नहीं रखतीं	69
--	----

अध्याय 21: क्षमता 1: आवश्यकता परिपक्वता प्रवाह	72
--	----

अध्याय 22: स्क्रम में RMF के साथ कार्य और सूचना का प्रवाह	75
---	----

अध्याय 23: RMF का प्रभाव	77
------------------------------------	----

अध्याय 24: RMF की ओर संक्रमण	78
--	----

अध्याय 25: यह आप पर निर्भर है	80
---	----

भाग VI: संसाधन	81
परिशिष्ट A: Scrum समस्या नहीं है	82
परिशिष्ट B: सिनैप्स फ्रेमवर्क™	84
परिशिष्ट C: RMF 1 के प्रति सामान्य आपत्तियां और बाधाएं	87
परिशिष्ट D: डीओडी प्रारंभिक मानदंड सूचियाँ	88
परिशिष्ट E: डीओआर स्टार्टर क्राइटेरिया सूचियाँ	89
सूची	90

इस पुस्तक के बारे में

रेडी एक ऐसी पुस्तक है जो सॉफ्टवेयर विकास में शामिल किसी भी व्यक्ति के लिए है जो कम डिलीवरी, लगातार दोबारा काम, और अस्पष्ट आवश्यकताओं से थक चुका है।

हो सकता है कि आपने पहले ही टीम के कार्यान्वयन कौशल में निवेश करने, अपने प्रक्रिया ढांचे के बेहतर क्रियान्वयन, या कोड को सुधारने का प्रयास किया हो और अभी भी अधिक सुधार की आवश्यकता हो।

ऐसा इसलिए है क्योंकि अधिकांश सॉफ्टवेयर विकास टीमों के लिए मुख्य बाधा टीम कौशल नहीं, बल्कि आवश्यकताओं की परिपक्वता है। सही कौशल वाली परिपक्व टीमें भी संघर्ष करती हैं जब वे अपरिपक्व आवश्यकताओं पर काम कर रही होती हैं।

रेडी आरएमएफ (आवश्यकता परिपक्वता प्रवाह) की शुरुआत करता है, जो आपकी मौजूदा प्रक्रिया को बदले बिना प्रोडक्ट और इंजीनियरिंग को संरेखित करने का एक व्यावहारिक और गहन संरचित दृष्टिकोण है।

चाहे आप स्क्रम, कानबान, या कुछ कस्टम का उपयोग कर रहे हों, आरएमएफ आपको कार्यक्षेत्र को स्थिर करने, अग्रेषण को समाप्त करने, और वास्तव में जो मायने रखता है उसे डिलीवर करने में मदद करता है।

यदि आपकी टीमें “लगभग पूरा हुआ” की स्थिति में फंसी हुई महसूस करती हैं, तो यह पुस्तक आपको इस चक्र को तोड़ने और अपनी टीम(टीमों) को हमेशा के लिए अनब्लॉक करने का तरीका दिखाएगी।

यह किसके लिए है

यह पुस्तक वास्तव में सॉफ्टवेयर विकास में शामिल किसी भी व्यक्ति के लिए है। इंजीनियर्स से लेकर प्रोडक्ट मैनेजर्स तक और व्यक्तिगत योगदानकर्ताओं से लेकर एग्जीक्यूटिव्स तक सभी के लिए।

यह पुस्तक आपके लिए है यदि आप सॉफ्टवेयर विकास में शामिल हैं और आपने देखा है कि जिस टीम के साथ या जिस पर आप काम करते हैं, उसमें निम्नलिखित में से एक या अधिक समस्याएं हैं:

- काम अक्सर एक पुनरावृत्ति से दूसरी में आगे बढ़ता है
- क्रियान्वयन टीमों को लगता है कि वे चलते लक्ष्यों को हिट करने की कोशिश कर रही हैं
- काम बहुत लंबे समय तक खुला रखा जाता है
- काम को बंद के रूप में चिह्नित कर दिया जाता है लेकिन चीजें वास्तव में पूरी नहीं होतीं
- किया गया काम अपेक्षाओं के अनुरूप नहीं होता
- काम नियमित रूप से बड़ी मात्रा में दोष उत्पन्न करता है
- बड़ी मात्रा में काम को नियमित रूप से दोबारा करना पड़ता है

यदि इनमें से कोई भी समस्या परिचित लगती है, तो रेडी मदद कर सकता है।

इस पुस्तक का उपयोग कैसे करें

यह पुस्तक व्यावहारिक होने के लिए डिज़ाइन की गई है। यह एक सैद्धांतिक निबंध या रणनीति डेक नहीं है - यह वास्तविक क्लाइंट कार्य पर आधारित और वास्तविक डिलीवरी दबाव के तहत फील्ड-टेस्टेड आरएमएफ (आवश्यकता परिपक्वता प्रवाह) को स्थापित करने के लिए एक हाउ-टू मैनुअल है।

अध्याय क्रम में लिखे गए हैं, लेकिन RMF स्वयं में मॉड्यूलर है। यह तीन मूल प्रथाओं से बना है:

- RMF 1: साझा समझ के लिए सहयोग
- RMF 2: कस्टम समाप्ति परिभाषाओं का उपयोग करके कार्य की पूर्णता को नियंत्रित करना
- RMF 3: कस्टम तत्परता परिभाषाओं का उपयोग करके कार्यान्वयन को नियंत्रित करना

प्रत्येक भाग, या जैसा कि हम उन्हें “आदत” कहते हैं, स्वयं में स्वतंत्र है, लेकिन वे एक-दूसरे पर निर्मित होते हैं। पुस्तक को इस तरह डिज़ाइन किया गया है कि आप उन्हें एक-एक करके, क्रम में अपना सकें। यह संरचना दर्शाती है कि हम टीमों को व्यवहार में RMF को कैसे अपनाने की सिफारिश करते हैं - प्रत्येक आदत को केवल पिछली आदत के काम करने के बाद ही जोड़ा जाता है।

यह टीमों को अभिभूत होने से बचाता है और प्रत्येक परिवर्तन को सफल होने का सर्वोत्तम अवसर देता है। आप इसे कैसे करें, इसके बारे में [अध्याय 9](#) में अधिक जानेंगे।

यदि आप मदद की तलाश में हैं - चाहे वह सलाह हो, कोचिंग हो, या आपकी नेतृत्व टीम से बात करने के लिए कोई हो - बेझिझक हमसे सीधे संपर्क करें।

और यदि आप RMF को लागू करने के लिए औपचारिक समर्थन की तलाश कर रहे हैं, तो Producore चरण-दर-चरण अपनाने के लिए डिज़ाइन किए गए कार्यक्रमों की एक पूरी श्रृंखला प्रदान करता है। आप <https://ready-book.link/rmf> पर अधिक जान सकते हैं।

लेखकों के बारे में

लुनिएल डी बीर आवश्यकता परिपक्वता प्रवाह (RMF) के निर्माता हैं, जो उत्पाद के इरादे और इंजीनियरिंग क्रियान्वयन के बीच की खाइयों को भरने के लिए एक व्यावहारिक प्रणाली है। उन्हें एजाइल परिवर्तनों का नेतृत्व करने, उत्पाद और इंजीनियरिंग को जोड़ने, और टीमों को स्पष्टता और आत्मविश्वास के साथ डिलीवर करने में मदद करने का 15 वर्षों से अधिक का अनुभव है।

लुनिएल ने Producore की क्षमता प्रबंधन प्रणाली भी विकसित की है, जो उत्पाद क्षमताओं को मॉडल करने के लिए एक अनुरेखणीय और स्केलेबल दृष्टिकोण है। उन्होंने पीकेबी-संचालित विकास (PKBDD) की कल्पना की, जो स्थायी उत्पाद आवश्यकताओं के प्रबंधन के लिए एक संस्करण-नियंत्रित प्रणाली है। ये उपकरण Producore में विकसित एक बड़े डिलीवरी ढांचे का हिस्सा हैं।

मैक्स गर्नसी, III एक सॉफ्टवेयर आर्किटेक्ट, शिक्षक और Producore के सह-संस्थापक हैं, जो संरचनात्मक और तकनीकी कठोरता के माध्यम से डिलीवरी की विफलताओं को ठीक करने के लिए समर्पित एक परामर्श कंपनी है। वस्तु-उन्मुख डिज़ाइन, रिक्वेस्टिंग, परीक्षण-संचालित विकास और डिज़ाइन पैटर्न में दो दशकों से अधिक के अनुभव के साथ, उन्होंने महत्वपूर्ण मिशन-क्रिटिकल सिस्टम डिलीवर किए हैं और बड़े पैमाने पर इंजीनियरिंग टीमों को कोचिंग प्रदान की है। उनका काम संगठनों को स्थायी डिलीवरी उत्कृष्टता प्राप्त करने में मदद करने के लिए गहन तकनीकी प्रथाओं को व्यावहारिक और प्रक्रिया परिवर्तन के साथ मिश्रित करता है।

Max ने PKBDD में महत्वपूर्ण योगदान दिया और व्यवहारात्मक विशिष्टता में अपनी गहरी विशेषज्ञता के माध्यम से Producore के व्यवहार-संचालित विकास (बीडीडी) के दृष्टिकोण का नेतृत्व किया।

साथ मिलकर, उनका कार्य स्पष्टता, पता लगाने की क्षमता, और गेटिंग को सॉफ्टवेयर डिलीवरी के लिए एक सुसंगत प्रणाली में एकीकृत करता है जो टीम के अभ्यास से लेकर संगठनात्मक क्षमता तक विस्तृत है।

प्रस्तावना

इंजीनियरिंग नेतृत्वकर्ताओं के लिए नोट

यदि आप किसी इंजीनियरिंग संगठन में वरिष्ठ नेतृत्वकर्ता हैं, तो संभवतः आपके पास प्रयास, अनुशासन, या प्रतिभाशाली लोगों की कमी नहीं है। फिर भी, किसी न किसी तरह, प्रोजेक्ट रुक जाते हैं। लक्ष्य फिसल जाते हैं। अपेक्षाएं पूरी नहीं होतीं। ऐसा इसलिए नहीं है क्योंकि आपकी टीम आलसी हैं - बल्कि इसलिए क्योंकि काम को परिभाषित करने, आकार देने और डिलीवर करने के तरीके में कुछ बुनियादी खामियां हैं।

यह पुस्तक एक नेतृत्व मार्गदर्शिका नहीं है। यह संरचनात्मक निदान के लिए एक उपकरण है। यह प्रकट करती है कि आपकी टीमों के भीतर वास्तव में क्या हो रहा है - क्यों “लगभग पूरा” बार-बार “अपूर्ण” में बदल जाता है, और क्यों स्थानीय प्रगति शायद ही कभी रणनीतिक परिणामों में बदलती है।

हो सकता है कि आप इन पन्नों में खुद को न देखें। लेकिन अगर आपकी टीम वह नहीं दे पा रही है जिसकी आपको जरूरत है, तो आप उन्हें देखेंगे। और जब आप ऐसा करेंगे, तो आपके पास अंततः इसे ठीक करने की भाषा - और प्रणाली - होगी।

लूनिंग की ओर से

सबसे पहले, यह पुस्तक मैक्स के बिना संभव नहीं होती, जिनकी धुंध और भूसे के बीच से देख पाने और किसी विचार को उसके सार में परिवर्तित करने की क्षमता मेरी समझ से परे है।

हम यहाँ कैसे पहुंचे?

अगर मैं पीछे देखूं, तो मुझे लगता है कि ऐसा इसलिए है क्योंकि मैं हमेशा से यह समझना चाहती थी कि चीजें वास्तव में कैसे काम करती हैं। चाहे वह धर्म हो, पोषण हो, या सॉफ्टवेयर डेवलपमेंट, मैं लगातार एक ही समस्या में उलझती रही: सही उत्तर जो दबाव में टिके नहीं रहते। इसलिए मैं खोजती रही - न केवल यह पूछती कि हम क्या करते हैं, बल्कि क्यों करते हैं, और जब यह काम नहीं करता तो क्या कमी रह जाती है।

प्रणाली में सबसे पहली दरारें एक ऐसी भूमिका में दिखाई दीं जहां मैं तीन टोपियां पहन रही थी: एक प्रसिद्ध टेक कंपनी में डेटा सेवाएं प्रदान करने वाली टीम के लिए Scrum Master, Product Owner, और Development Manager (!!)। हम वही कर रहे थे जो Scrum कहता था - छोटे Sprints, बैकलॉग में स्टोरीज, आधे दिन में प्लानिंग - लेकिन हर बार जब हम एक नया Sprint शुरू करते, हम घर्षण का सामना करते। टीम समस्या को पूरी तरह से नहीं समझती थी, हमें Sprint के बीच में आवश्यकताओं को दोबारा देखना और संशोधित करना पड़ता था, रोकी जा सकने वाली निर्भरताएं सामने आकर हमें देरी करवातीं, और महत्वपूर्ण कदम छूट जाते थे।

तो मैंने कुछ अलग करना शुरू किया। मैं हर स्टोरी के लिए टीम और हितधारकों को एक कमरे में बुलाती, विस्तार से समस्या पर चर्चा करती, एक साथ मिलकर समाधान पर विचार-मंथन करती, और तभी स्टोरी लिखती। Sprint प्लानिंग घटकर एक घंटे की रह गई, और हमारी डिलीवरी सफलता में जबरदस्त वृद्धि हुई।

धीरे-धीरे, मैं समझने लगी कि सफलता Sprint के भीतर कड़ी मेहनत से नहीं आती। यह उस संरचना से आती है जो आप इसके शुरू होने से पहले स्थापित करते हैं।

बाद में, Jeff Sutherland को “तैयारी की परिभाषाएं” के बारे में बात करते हुए सुनने के बाद, मुझे पता था कि इसमें कुछ मूल्यवान था—लेकिन यह पर्याप्त नहीं था। आवश्यकताओं, UX, UI, शोध, और बाद में BDD के साथ मेरे अनुभव ने मुझे दिखाया कि अलग-अलग कार्य मर्जों के लिए अलग-अलग प्रकार की तैयारी की आवश्यकता होती है। कुछ को व्यवहार विनिर्देश की जरूरत होती है। कुछ को सिस्टम एक्सेस की जरूरत होती है। कुछ को पूर्ण क्षमता अनुरेखण की आवश्यकता होती है।

और इन सभी को साझा समझ की आवश्यकता होती है जो वास्तव में पुष्टि की गई हो—न कि मान ली गई हो।

जैसे-जैसे मैंने और टीमों के साथ काम किया, मैंने हर जगह एक ही पैटर्न देखा: छूटे हुए कदम, अपूर्ण निर्भरताएं, टीमों अपना सर्वश्रेष्ठ प्रयास कर रही थीं लेकिन लगातार उन समस्याओं को ठीक करने के लिए संघर्ष कर रही थीं जिन्हें रोका जा सकता था। यहां तक कि महान टीमों भी संघर्ष करती थीं—इसलिए नहीं कि वे कमजोर थीं, बल्कि इसलिए कि उनके पास एक ऐसी संरचना की कमी थी जो तैयारी को स्पष्ट करती।

उस सभी सीख, पुनरावृत्ति और निराशा का परिणाम तैयारी के प्रबंधन के लिए एक संरचित प्रणाली है।

यही इस पुस्तक का विषय है।

मुझे आशा है कि यह आपको वास्तविक समस्याओं और उन्हें कैसे ठीक किया जाए, इस बारे में स्पष्टता प्रदान करेगी। मुझे आशा है कि यह आपको उन प्रथाओं का बचाव करने के लिए भाषा प्रदान करेगी जो “अतिरिक्त” लग सकती हैं लेकिन वास्तव में आवश्यक हैं।

और सबसे महत्वपूर्ण, मुझे आशा है कि यह टीमों को कम तनाव, कम आश्चर्य, और बेहतर परिणामों के साथ डिलीवर करने में मदद करेगी।

यदि हम इसे सही करते हैं, तो हम उद्योग के अरबों डॉलर बचाएंगे।

लेकिन इससे भी महत्वपूर्ण, हम लोगों को उनका मानसिक संतुलन वापस देंगे।

Max से

मैं दशकों से विभिन्न कोणों से इस समस्या पर काम कर रहा हूं, लेकिन Luniel से मिलने से पहले मेरी प्रगति रुक गई थी।

ऐसा इसलिए था क्योंकि, उन्हें जानने से पहले, मैं मूल रूप से समस्या को एक तकनीकी समस्या के रूप में देख रहा था। मैं टीमों को Test-Driven Development (TDD), रिफैक्टरिंग, उन्नत सॉफ्टवेयर डिज़ाइन और बाद में Acceptance-Test-Driven Development (ATDD) या Behavior-Driven Development (BDD) जैसी चीजों को अपनाने में मदद करने पर केंद्रित था।

उन अधिकांश मामलों में, इस पुस्तक में संबोधित समस्या को उन तकनीकी प्रथाओं को स्थापित करने के एक कार्यान्वयन विवरण के रूप में माना जाता था।

इसका यह मतलब नहीं है कि मैं अब तकनीकी प्रथाओं को महत्व नहीं देता। मैं अभी भी मानता हूं कि वे गहराई से महत्वपूर्ण हैं, लेकिन वे सॉफ्टवेयर विकास में तैयारी की समस्या का सीधे समाधान नहीं करतीं। इसके बजाय, वे उस समस्या को सतह पर लाती हैं और फिर लोग अपनी प्रक्रिया में एक पैच लगा देते हैं ताकि उसे “बस इतना” संबोधित किया जा सके जो वे लागू करने की कोशिश कर रहे तकनीकी प्रथाओं को समर्थन करे।

मैं यह भी संबोधित करना चाहता हूं कि यह पुस्तक किसकी मदद कर सकती है। संक्षिप्त उत्तर है “शायद सॉफ्टवेयर विकास में लगभग हर कोई”, लेकिन वास्तविक उत्तर में ऐसी बारीकियां हैं जो इसे मूल अर्थ को बदले बिना विभिन्न वातावरणों के अनुरूप बनाने में मदद करती हैं।

ऐसी टीमों हैं जिन्हें इस पुस्तक में दी गई समाधान की आवश्यकता है। आप उनमें से एक का सरलीकृत संस्करण अध्याय 1 में देखेंगे।

ऐसी टीमों भी हैं जिन्हें सख्ती से इस तरह की प्रणाली की आवश्यकता नहीं है जैसी हम प्रस्तावित कर रहे हैं, लेकिन फिर भी इससे लाभान्वित हो सकती हैं।

मैंने जिस सर्वश्रेष्ठ टीम के साथ काम किया—जो आसानी से अगली सर्वश्रेष्ठ टीम से एक पूर्ण मानक विचलन इकाई ऊपर थी, यदि दो नहीं—वह सेंट्रल ओरेगन के पिछड़े क्षेत्र में

स्थित थी। वे इतनी उच्च-प्रदर्शन करने वाली टीम थी कि वे अपनी क्षमता के बल पर ऐसी प्रणाली की अनुपस्थिति को पार कर सकती थी। फिर भी, उस समय मेरे प्रबंधक, टॉम बरेरास, ने मुझसे कुछ इस तरह कहा था “मैंने देखा है कि जब हम शुरुआत में परीक्षणों के बारे में कुछ समय बात करते हैं तो हमारी कार्य-कथाएँ बेहतर होती हैं।”

यह, फिर से, कुछ ऐसा था जिसे मैंने उस समय परीक्षण-विकास और तकनीकी क्रियान्वयन के दृष्टिकोण से देखा, लेकिन अब मैं जानता हूँ कि यह एक और संकेत था कि तैयारी एक ऐसा कारक था जो टीम को प्रभावित कर रहा था... वह विशेष टीम इतनी सक्षम और प्रतिक्रिया में तेज थी कि वे बाधाओं को पहले से रोकने के बजाय उनके होने पर प्रतिक्रिया करके भी सफल हो सकती थी।

यदि आप उस तरह के व्यक्ति हैं जिन्हें तैयारी के बारे में सख्ती से चिंता करने की आवश्यकता नहीं है क्योंकि आप इससे पार पा सकते हैं, या आप ऐसी ही टीम के साथ काम करते हैं, तब भी आप इस पुस्तक की सामग्री से लाभान्वित हो सकते हैं।

भाग I: कुछ तो कम है

जब वही चीज़ें बेहतर करने से मदद नहीं मिलती, तब देखिए कि क्या नहीं किया जा रहा है।

अध्याय 1: छिपी हुई समस्या

यह एक सच्ची¹ कहानी है एक बैंक की। हम इसे बस “द बैंक” कहेंगे। यह एक प्रकार की संघीय क्रेडिट संस्था है जो संयुक्त राज्य अमेरिका के राष्ट्रीय वित्तीय बुनियादी ढांचे का हिस्सा है।

हमें (लुनिएल और मैक्स को) द बैंक में इसलिए बुलाया गया क्योंकि वह एक सॉफ्टवेयर प्रोजेक्ट को पूरा करने में संघर्ष कर रहा था। यह हमने देखे गए सबसे अव्यवस्थित वातावरणों में से एक था, और यही कारण है कि हमने इसे प्रारंभिक केस स्टडी के रूप में चुना: अगर द बैंक में सार्थक परिवर्तन संभव था, तो यह कहीं भी संभव है।

प्रोजेक्ट पर एक त्वरित नोट

जब हम इस पुस्तक में “प्रोजेक्ट” शब्द का उपयोग करते हैं, तो हम इसका अर्थ प्रोजेक्ट मैनेजमेंट के संदर्भ में लेते हैं। हालांकि इस शब्द के अर्थ के बारे में विभिन्न विचार हैं, हम प्रोजेक्ट मैनेजमेंट इंस्टीट्यूट की परिभाषा का उपयोग कर रहे हैं:

“एक प्रोजेक्ट एक अस्थायी प्रयास है जो एक अनूठा उत्पाद, सेवा, या परिणाम बनाने के लिए किया जाता है।”

इसका मतलब है कि एक प्रोजेक्ट की एक निश्चित शुरुआत और अंत होता है। जब एक प्रोजेक्ट बंद होता है, प्रोजेक्ट ज्ञान और कलाकृतियों को संग्रहीत किया जाता है, टीम के सदस्यों को मुक्त किया जाता है, और अनुबंधों को अंतिम रूप दिया जाता है।

इस पुस्तक में, एक प्रोजेक्ट मूल रूप से क्रियान्वयन के बारे में है। PMI द्वारा परिभाषित अधिकांश प्रोजेक्ट व्यवहार्यता या डिजाइन से शुरू होते हैं। प्रोजेक्ट शुरू होने तक विज़न और रणनीति पहले ही हो चुकी होती है।

एक प्रोजेक्ट उस विज़न और रणनीति से जन्म लेता है, और यह इस बात पर सफल या असफल होता है कि क्या परिकल्पित लक्ष्य प्राप्त किए जाते हैं—न कि इस बात पर कि क्या वे लक्ष्य सही थे।

¹हमने संबंधित व्यक्तियों और संस्थानों की गोपनीयता की सुरक्षा के लिए पहचान संबंधी विवरणों को बदल दिया है।

आप “प्रोजेक्ट” शब्द का अलग तरह से उपयोग कर सकते हैं, और यह ठीक है। बस जान लें कि जब हम इसका उपयोग करते हैं, तो हम ऊपर दी गई परिभाषा और संदर्भ का उल्लेख कर रहे हैं।

इनमें से कुछ भी यह नहीं दर्शाता कि हम सॉफ्टवेयर विकास के लिए प्रोजेक्ट मैनेजमेंट के उपयोग का समर्थन करते हैं। बिल्कुल विपरीत। लेकिन हम मानते हैं कि फिर भी इसका उपयोग किया जाता है। हम इस समस्या को बाद में [अध्याय 6](#) में संबोधित करते हैं।

1.1: क्लासिक रीराइट

द बैंक कई कारणों से अपने ऋण पुनर्भुगतान पोर्टल को फिर से लिख रहा था।

पुरानी प्रणाली, जो पूरी तरह से C#/.NET समाधान थी, में बग्स थे। ग्राहक संतुष्टि को कम करने के अलावा, यह बहुत महंगी सहायता घटनाओं का एक अनवरत प्रवाह भी उत्पन्न करती थी जिसमें किसी को सिस्टम द्वारा की गई त्रुटि को सुधारने के लिए मैनुअल रूप से डेटाबेस में हेरफेर करना पड़ता था।

पुरानी प्रणाली रखरखाव की दृष्टि से भी जीर्ण थी। इंजीनियरों के लिए सार्थक परिवर्तन करना लगभग असंभव था, और यहां तक कि जब वे कर भी सकते थे, तो यह एक बेहद जोखिम भरा प्रस्ताव था।

रीराइट को यह सब बदलना था।

नई प्रणाली में अभी भी C#/.NET बैकएंड होना था, लेकिन वह पूरी तरह से टेस्ट से कवर किया जाना था। फ्रंटएंड को OutSystems में कार्यान्वित किया जाना था, जो एक लोकप्रिय लो- या नो-कोड समाधान है जो किसी संगठन को एक ही स्थान पर एप्लिकेशन को परिभाषित करने और जब भी वे अपने परिवर्तनों को प्रकाशित करने का निर्णय लें, स्वचालित रूप से एक वेब ऐप, एक एंड्रॉइड ऐप, और एक iOS ऐप प्राप्त करने की अनुमति देता है।

आशा थी कि नई प्रणाली त्रुटि-मुक्त होगी, जो ग्राहक संतुष्टि में सुधार करेगी और सहायता लागत को महत्वपूर्ण रूप से कम करेगी।

उन्हें यह भी आशा थी कि रीराइट डेवलपर्स को अनब्लॉक कर देगा—बैकएंड के प्रति एक अधिक अनुशासित दृष्टिकोण और फ्रंटएंड के लिए लो-कोड दृष्टिकोण के संयोजन से नई सुविधाओं की लागत और जोखिम काफी कम हो जाएगा।

OutSystems में जाने का एक अच्छा साइड इफेक्ट यह था कि उन्हें दोनों प्रमुख प्लेटफॉर्म पर एक साफ, आधुनिक मोबाइल ऐप मिल जाएगा।

यह सपना था जब उन्होंने इस कहानी की शुरुआत से तीन साल पहले शुरू किया था। वास्तविकता यह थी कि, अब तक, टीमों ने कुछ भी शिप नहीं किया था।

1.2: समस्या पर विभिन्न दृष्टिकोण

जब हमने कार्यकारी नेतृत्व टीम से बात की, तो हमने इस तथ्य पर बहुत स्वाभाविक निराशा सुनी कि उन्होंने इतना बड़ा निवेश किया था लेकिन बिल्कुल भी रणनीतिक प्रगति नहीं हुई थी।

उनके दृष्टिकोण से, उन्होंने सब कुछ आजमाया था। उन्होंने स्टाफ बदला, स्टाफ बढ़ाया, बजट बदला, दबाव बढ़ाया, और सलाहकारों की एक परेड लाई (जिसमें यह मजबूती से संकेत किया गया था कि हम अंतिम थे)। कुछ भी बेहतर नहीं लगा—कम से कम ऐसे तरीके से नहीं जिसे वे माप सकें, क्योंकि वे जो देखते थे वह यही था कि “सूई” एक तिमाही में शून्य पर थी और फिर अगली तिमाही में भी शून्य पर थी।

वे और “अदृश्य प्रगति” नहीं चाहते थे। वे परिणाम चाहते थे।

जब हमने उत्पाद संगठन के प्रबंधन से बात की, तो हमें एक थोड़ी अलग (लेकिन फिर भी समान) कहानी मिली क्योंकि वे इंजीनियरिंग के साथ अधिक सीधे काम करते थे।

उनके लिए ऐसा नहीं था कि टीमों ने कुछ नहीं किया। बल्कि यह था कि टीमों कभी वह नहीं करतीं जो उनसे करने को कहा जाता था। यह लगभग गारंटी थी: चाहे कितना भी सरल काम हो और चाहे कितनी भी स्पष्टता से इसे बताया गया हो, जब टीमों ने जो बनाया था उसका मूल्यांकन करने का समय आता, तो आपको कुछ पूरी तरह से अलग ही मिलता।

स्थिति यहाँ तक पहुँच गई थी कि एक चलता हुआ मजाक बन गया था कि “हमें यह पता लगाना होगा कि हम क्या नहीं चाहते, ताकि कम से कम हमें वह मिल सके जो हम वास्तव में चाहते हैं।”

इंजीनियरिंग कार्यकारियों ने चीजों को बिल्कुल अलग तरीके से देखा।

उनके लिए, मुद्दा यह था कि प्रोडक्ट कार्यान्वयन योग्य आवश्यकताएं नहीं दे रहा था और प्रोडक्ट पर्याप्त आवश्यकताएं नहीं दे रहा था। अगर प्रोडक्ट बस “कार्यक्रम के साथ चल सकता”, तो टीमों समय पर और बजट के भीतर वह सब कुछ दे पाती जो वे चाहते थे।

उन्होंने कोड लिखने और डिलीवर करने के तरीके को आधुनिक बनाने में गंभीर निवेश किया था और, उनकी नजर में, प्रोडक्ट स्पष्ट आवश्यकताएं नहीं दे रहा था।

जब हमने अन्य सलाहकारों से बात की (जिन्होंने हमें संगठन में भेजा था), उन्होंने सही ढंग से उस खराबी पर ध्यान केंद्रित किया जो वे देख रहे थे: हर कोई किसी और को दोष देने पर बहुत ध्यान केंद्रित था। उन्होंने हमें शुरू में इसलिए बुलाया क्योंकि वे स्टाफिंग रणनीति को लेकर चिंतित थे और व्यक्तिगत योगदानकर्ताओं का मूल्यांकन चाहते थे, लेकिन उन्होंने कार्यकारी स्तर पर आरोप लगाने और कार्य-मास्टर की मानसिकता को मुख्य समस्या के रूप में देखा।

1.3: हमारी जाँच

हमारा प्रारंभिक चार्टर टीमों का मूल्यांकन करना और जरूरत पड़ने पर उनके कौशल को बेहतर बनाने में मदद करना था, इसलिए हमने फ्रंट लाइन पर काम करने वाले लोगों की जाँच शुरू की।

निश्चित रूप से सुधार की गुंजाइश थी।

प्रोडक्ट साइड पर व्यक्तिगत योगदानकर्ताओं के पास वास्तव में आवश्यक कौशल नहीं थे। वास्तविकता में, वे ज्यादातर प्रोजेक्ट मैनेजर थे जिन्हें प्रोडक्ट ओनर (पीओ) या प्रोडक्ट मैनेजर की भूमिका में धकेल दिया गया था।

नतीजतन, आधे लोग “अस्पष्ट” आवश्यकताएं लिखते थे और फिर बिना किसी आलोचनात्मक विश्लेषण के उस इटिरेशन में टीमों ने जो किया उसे (शाब्दिक रूप से) स्वीकार कर लेते थे। बाकी आधे लोग उसी तरह की आवश्यकताएं लिखते थे और फिर दावा करते थे कि टीमों को वह सब “पता होना चाहिए था” जिसके बारे में उन्होंने कभी बात ही नहीं की और काम के आइटम्स को अनिश्चित काल तक खुला रखते थे।

बैंक अपने काम के बैकलॉग को प्रबंधित और ट्रैक करने के लिए इस्तेमाल करता था। हालांकि इस पुस्तक में हम जो कवर करते हैं वह ज्यादातर स्क्रम से अलग है, हम पूरी किताब में स्क्रम शब्दावली का उपयोग करते हैं क्योंकि अधिकांश—या कम से कम बहुसंख्यक—टीमें स्क्रम का उपयोग करती हैं।



परिभाषा: स्क्रम

स्क्रम एक हल्का ढांचा है जो लोगों, टीमों और संगठनों को जटिल समस्याओं के लिए अनुकूली समाधानों के माध्यम से मूल्य उत्पन्न करने में मदद करता है। संक्षेप में:

1. एक प्रोडक्ट ओनर जटिल समस्या के काम को प्रोडक्ट बैकलॉग में व्यवस्थित करता है
2. स्क्रम टीम एक स्प्रिंट के दौरान काम के एक चयन को मूल्य के इंक्रिमेंट में बदलती है
3. स्क्रम टीम और उसके हितधारक परिणामों का निरीक्षण करते हैं और अगले स्प्रिंट के लिए समायोजित करते हैं
4. दोहराएं

यदि आप स्क्रम और उसकी शब्दावली से अपरिचित हैं, तो हम [स्क्रम गाइड](#) के 2020 संस्करण की समीक्षा करने की सलाह देते हैं। यह एक त्वरित, ज्ञानवर्धक पाठ है।

इसी तरह, हमने पाया कि तकनीकी टीमों कोडिंग कौशल के मामले में सामान्य स्तर से काफी नीचे थीं (-2σ, अधिकतम) और इसके ऊपर से परिवर्तन के प्रति अत्यधिक प्रतिरोधी थीं। स्वाभाविक परिणाम के रूप में, कोड की गुणवत्ता बेहद खराब थी।

फिर भी, टीमों के अनुसार, यह वजह नहीं थी कि वे डिलीवरी नहीं कर पा रहे थे। उनके लिए, यह अस्पष्ट आवश्यकताएं और प्रोडक्ट द्वारा मध्य-स्प्रिंट परिवर्तन थे जो प्रोजेक्ट को मार रहे थे।

...और कोई भी बड़ी समस्या के बारे में बात भी नहीं कर रहा था, जो इतनी बेतुकी है कि जब तक आप इसे जी न लें, यह मनगढ़ंत लगती है।

इंजीनियरिंग टीमों की एक आदत थी कि वे किसी आवश्यकता को न समझें, कुछ यादृच्छिक बना दें, और फिर “कार्य आइटम पूरा करने” का श्रेय मांगें।

हम छोटी गलतफहमी की बात नहीं कर रहे हैं। हम पूर्ण असंबद्धता की बात कर रहे हैं: हम उन्हें कुछ विशेष परिस्थितियों में मूलधन पर फंड के आवेदन को अक्षम करने के लिए कहते, और वे इसके बजाय द्वितीयक पुष्टिकरण ईमेल पता जोड़ने की क्षमता को अक्षम कर देते।

फिर वे हमें बताते कि यही है जो हमने मांगा था।

1.4: गहराई में जाना

दोनों कौशल अंतराल को संबोधित किया जा सकता था, लेकिन हमें संदेह था कि वे वास्तविक बाधाएं थीं।

कुछ और गड़बड़ था, इसलिए हमने और गहराई से जांच की। हमने इस प्रश्न से शुरुआत की: आवश्यकताएं लिखने में इतना समय क्यों लगता था और इतने खराब परिणाम क्यों मिलते थे?

एक कारण यह था कि सार्थक आवश्यकता लिखने के लिए आवश्यक ज्ञान बेहद कम मात्रा में उपलब्ध था। इसका थोड़ा हिस्सा इंजीनियरिंग और प्रोडक्ट टीमों के पास था।

इसका कुछ हिस्सा पुरानी प्रणाली के कोड में समाहित था। कुछ पूरी तरह से खो गया था। हालांकि, अधिकांश बैंक की विभिन्न इकाइयों में फैले विषय विशेषज्ञों के पास सामूहिक ज्ञान के रूप में संग्रहीत था। इसका मतलब है कि एक ऐसी आवश्यकता का निर्माण जो वास्तव में रणनीतिक लक्ष्य को आगे बढ़ाती है, एक अत्यंत श्रम- और समय-गहन गतिविधि थी।

इसके विपरीत था फीचर-भूखे नेतृत्व टीम की कार्यक्षमता के लिए अतृप्त भूख। आदेश था “इंजीनियरों को काम में व्यस्त रखो—उन्हें आवश्यकताओं से भर दो”। ध्यान टीमों को व्यस्त रखने के लिए आवश्यकताओं की मात्रा पर था—एक दृष्टिकोण जो उस देखभाल और समय के विपरीत था जो एक ऐसी आवश्यकता को परिभाषित करने के लिए जरूरी था जो “सुधार ला सके”।

1.5: चीजों को बेहतर बनाना बेहतर नहीं बना

ये सभी समस्याएं समाधान योग्य थीं, फिर भी उन्हें संबोधित करने से मदद नहीं मिली।

सॉफ्टवेयर विकास तकनीकों में पिछले सुधारों ने टीमों को डिलीवर करने में मदद नहीं की थी, लेकिन मैक्स ने टीमों को और सुधार में मदद करने का प्रयास किया।

उन्होंने मध्य-बीसवीं सदी के प्रोग्रामिंग सिद्धांतों से क्रांतिकारी अवधारणाएं पेश कीं, जैसे “उस कोड को 27² बार कॉपी और पेस्ट मत करो, इसे एक फंक्शन में डालो और उसे कॉल करो”। यह सुझाव अकेले ही नए कोड की गुणवत्ता में नाटकीय सुधार लाया और उन्हें गुणवत्ता में सुधार करने में मदद की।

²यह न केवल अतिशयोक्ति नहीं है, बल्कि यह सबसे खराब स्थिति भी नहीं है। एक मामले में, एक ही एल्गोरिथ्म की लगभग सौ बिल्कुल समान प्रतिलिपियाँ थीं।

वह और अन्य बुनियादी कोडिंग सलाह ने उन्हें बेहतर कोड लिखने में मदद की जिसे वे भविष्य में आसानी से बनाए रख सकते थे।

...लेकिन इससे प्रोजेक्ट को आगे बढ़ाने में मदद नहीं मिली।

प्रोडक्ट की तरफ से, लुनिएल BDD को पेश कर सके और यह सुनिश्चित किया कि प्रोडक्ट ओनर्स आवश्यकताओं की पूरी तरह से जांच करें, इससे पहले कि वे टीमों को सौंपी जाएं।

उन्होंने टीमों को सहयोग करने के लिए प्रेरित किया और उनका उपयोग यह मूल्यांकन करने के लिए किया कि क्या कोई प्रोडक्ट बैकलॉग आइटम (PBI) वास्तव में पूरा हुआ है या नहीं।



परिभाषा: प्रोडक्ट बैकलॉग आइटम (PBI)

एक प्रोडक्ट बैकलॉग आइटम (PBI) प्रोडक्ट बैकलॉग में काम की एक विवेकपूर्ण इकाई है जो उत्पाद में संभावित परिवर्तन, जोड़ या सुधार का प्रतिनिधित्व करती है। PBI कई रूप ले सकते हैं - फीचर, बग फिक्स, तकनीकी सुधार, शोध कार्य, आदि - और ये उत्पाद मूल्य में अपने योगदान से परिभाषित होते हैं।

कई टीमों PBI को “स्टोरीज” या “यूजर स्टोरीज” के रूप में संदर्भित करती हैं, लेकिन सही स्क्रम शब्द “प्रोडक्ट बैकलॉग आइटम” या “PBI” है। एक बार जब PBI को स्प्रिंट में प्रतिबद्ध कर दिया जाता है, तो यह स्प्रिंट बैकलॉग का भी हिस्सा बन जाता है। सरलता और तटस्थता के लिए, हम “प्रोडक्ट बैकलॉग आइटम” या “PBI” का उपयोग किसी भी कार्य आइटम को संदर्भित करने के लिए करते हैं जिसे स्क्रम टीम प्रबंधित कर रही है - चाहे आप इसे प्रोडक्ट बैकलॉग आइटम, स्प्रिंट बैकलॉग आइटम (SBI), यूजर स्टोरी, स्टोरी, वर्क आइटम, या बैकलॉग आइटम कहें।

इसने स्पष्टता जोड़ी, लेकिन इसने प्रवाह नहीं बनाया।

हमें लाने वाले पार्टनर सलाहकारों की मदद से, हम टीमों और आवश्यकता लेखकों पर पड़ने वाले बिल्कुल दबाव को (अस्थायी रूप से) कम करने में सक्षम थे।

यह थोड़ा विश्वास बनाने में मदद कर सकता था, लेकिन इससे कोई ठोस परिणाम नहीं मिला।

हमने एक ज्ञान-आधार बनाना भी शुरू किया जो लोगों को आवश्यकताएं लिखने के लिए आवश्यक व्यावसायिक ज्ञान को ट्रैक करने में मदद करता था और उन स्थानों की पहचान करता था जहां उस ज्ञान में अंतराल थे।

यह आवश्यकताएं लिखने की प्रक्रिया को तेज कर दिया, लेकिन इससे उत्पाद बाजार में नहीं पहुंच पाया।

महीनों की बातचीत के बाद, हमने नेतृत्व को यह दिखा दिया था कि वे कहाँ खड़े हैं, लेकिन वे अपने लक्ष्यों से काफी दूर थे। और वे करीब भी नहीं आ रहे थे।

वे अपनी पुरानी रणनीति पर वापस लौटने की तैयारी कर रहे थे - टीमों को “काम से लाद देना” ताकि वे हमेशा व्यस्त रहें।

1.6: निष्कासन की प्रक्रिया

यह कहना आसान होता कि “यह निराशाजनक है”। कई बहाने बनाए जा सकते थे:

- इंजीनियरिंग टीम कम कुशल थी (वह थी)
- आवश्यकताएं लिखने वालों के पास सही कौशल नहीं था (उनके पास नहीं था)
- नेतृत्व अवास्तविक अपेक्षाओं से टीमों को कुचल रहा था (वे कर रहे थे)
- संगठन में कार्य करने के लिए आवश्यक महत्वपूर्ण व्यावसायिक ज्ञान की कमी थी (यह था)
- कार्यकारी एक-दूसरे पर विश्वास नहीं करते प्रतीत होते थे (वे नहीं करते थे)

ये सभी बातें सच थीं। फिर भी इन सभी क्षेत्रों में सुधार किया गया था और इनमें से किसी ने भी मूल समस्या को बेहतर नहीं बनाया। इनमें से किसी ने भी इंजीनियरिंग टीमों, प्रोडक्ट, प्रबंधन, या कार्यकारी वर्ग को अपने लक्ष्यों के करीब नहीं पहुंचाया।

...और यहीं समाधान का संकेत छिपा है: पहले सूचीबद्ध की गई सभी समस्याएं वे थीं जो लोग पहले से ही देख सकते थे।



यदि जो चर आप देख सकते हैं वे कोई अंतर नहीं लाते, तो जरूर कोई ऐसा चर है जिसे आप नहीं देख रहे हैं जो प्रभाव डालता है।

वास्तविक मुद्दा वह समस्या थी जिसके बारे में किसी को पता भी नहीं था।

1.7: वास्तविक दोषी की खोज

बैंक को अपने लक्ष्यों को प्राप्त करने से रोकने वाले वास्तविक दोषी - वास्तविक कारण - की खोज में, हमें कहीं से शुरुआत करनी थी।

एक उचित जगह यह देखना था कि असफल हुए पीबीआई (अधिकांश) में क्या समान था।

हमने उन चीजों को हटाकर शुरुआत की जिनके बारे में हम जानते थे कि वे समान नहीं थीं क्योंकि वे काफी भिन्न थीं:

- सिस्टम का कौन सा हिस्सा: कुछ पीबीआई केवल बैकएंड को प्रभावित करते थे, अन्य केवल फ्रंटएंड को, और कुछ दोनों भागों को प्रभावित करते थे
- किस टीम ने काम किया - यह मायने नहीं रखता था कि काम किसने किया, असफल होने की संभावना अधिक थी
- किस पीओ ने काम लिखा - टीमों की तरह ही स्थिति थी

फिर हमने उन चीजों को देखना शुरू किया जो समान थीं। सूची न तो बहुत लंबी थी, और न ही बहुत छोटी:

- इंजीनियरिंग टीमें
- प्रोडक्ट ओनर्स
- नेतृत्व
- संस्कृति
- डेवलपमेंट एनवायरनमेंट
- इंजीनियरिंग प्रथाएं
- आवश्यकताओं को लिखने की तकनीक
- डोमेन (वित्त)
- निर्भरता सेवाएं

इनमें से कई को भी तुरंत खारिज किया जा सकता था। टीमों, प्रोडक्ट ओनर्स, नेतृत्व, संस्कृति और डेवलपमेंट एनवायरनमेंट में हाल ही में सुधार किया गया था, लेकिन इसका वास्तविक आउटपुट पर कोई प्रभाव नहीं पड़ा। हमने व्यक्तिगत रूप से इंजीनियरिंग और आवश्यकताओं को लिखने की प्रथाओं में सुधार में मदद की थी और यह पुष्टि की थी कि वे सुधार बने रहे, लेकिन फिर भी यह मदद नहीं कर रहा था।

आप डोमेन को दोष नहीं दे सकते। वित्त रिकॉर्ड किए गए इतिहास में की जाने वाली सबसे पुरानी गणनाओं में से एक है। यह बेहद परिपक्व है। इसके अलावा, अन्य बैंक सॉफ्टवेयर

को डिप्लॉय कर रहे थे, जो इस (स्पष्ट रूप से अतिरंजित) परिकल्पना को गलत साबित कर रहे थे कि बैंक ऐसा नहीं कर सकते।

निर्भरता सेवाओं को भी दोष नहीं दिया जा सकता था, क्योंकि उन्हें भी उतनी ही परेशानी हो रही थी जितनी हम जिस पहल को देख रहे थे...

...लेकिन इससे हमें सोचने पर मजबूर किया: क्या होगा अगर हम विफलता के कारणों का विश्लेषण करना शुरू करें?

1.8: विफलता के बीजों का विश्लेषण

एक PBI उत्पाद को आगे बढ़ाने में विफल रहा क्योंकि टीम ने, जैसा कि वे करने के आदी थे, कुछ पूरी तरह से यादृच्छिक और लगभग पूरी तरह से अनुरोध से असंबंधित काम किया। यह स्पष्ट रूप से एक संकेत है कि उन्होंने कार्य आइटम को नहीं समझा। इसलिए समझ एक बड़ा उम्मीदवार था, भले ही हमने उन्हें BDD अपनाने में मदद करते समय इस पर काम किया था।

एक अन्य PBI विफल रहा क्योंकि उन्होंने गणनाएं गलत कीं। यह एक और सबूत है कि समझ मुख्य मुद्दा हो सकता है।

एक तीसरा कार्य आइटम जिसका हमने विश्लेषण किया, वह वास्तव में प्रोडक्ट ओनर द्वारा ठीक से लागू नहीं किया गया था - उसने टीम के कहने पर बिना सोचे-समझे इसे बंद कर दिया। इसने हमारी परिकल्पना को कुछ तनाव दिया, लेकिन फिर भी यह तर्क दिया जा सकता था कि वह नहीं समझती थी कि कार्य आइटम उच्च स्तरीय योजना में कैसे फिट होता है।

शायद। कुछ हद तक। अगर हम इसे इस तरह से देखते समय बहुत कड़ी नजर से देखते।

फिर हमें एक ऐसा PBI मिला जो बिल्कुल भी इस पैटर्न में फिट नहीं होता था। टीम समझती प्रतीत होती थी - हालांकि यह सत्यापित करने का कोई तरीका नहीं है कि वे वास्तव में समझे या नहीं। लेकिन इससे कोई फर्क नहीं पड़ता: उन्हें अपने दम पर सफल या विफल होने का मौका ही नहीं मिला क्योंकि वे एक निर्भरता में फंस गए जिसे अपडेट करने की आवश्यकता थी और उन्हें अपना काम कई Sprints के लिए स्थगित करना पड़ा।

भले ही उन्हें यह नहीं समझ में आया था कि उन्हें क्या करना था, उस बैकलॉग आइटम के साथ उनके पास कोई मौका ही नहीं था, इसलिए उस मामले में समझ **समस्या नहीं** थी।

बेशक, एक अपवाद किसी विशेष मूल कारण का खंडन नहीं करता, लेकिन इसने हमारी जिज्ञासा को जगा दिया। हम अन्य खंडनकारी साक्ष्य खोजने लगे।

और हमें वे मिल गए। ऐसे कार्य आइटम थे जो:

- इसलिए असफल हुए क्योंकि टीम जानती थी कि उसे समझ नहीं आ रहा है, लेकिन समस्या को सुलझाने के लिए कोई विषय विशेषज्ञ नहीं मिल सका
- अपस्ट्रीम सर्विसेज के कार्य करने के तरीके के कारण एक वर्कअराउंड के रूप में उपयोगकर्ता के लिए खराब अनुभव में बदल गए
- अपस्ट्रीम निर्भरताएं तैयार न होने के कारण स्थगित करना पड़े
- पूरे नहीं किए जा सके क्योंकि परीक्षक समय पर टेस्ट डेटा एकत्र नहीं कर पाए
- बंद कर दिए गए लेकिन फिर से करने पड़े क्योंकि मांग ही गलत थी
- इसलिए असफल हुए क्योंकि टीम को यह एहसास नहीं था कि मौजूदा कोड पहले से ही कितना जटिल था
- जिनका अनुमान ही नहीं लगाया गया
- जिनका बहुत कम अनुमान लगाया गया³
- स्प्रींट के बीच में बदल गए क्योंकि पीओ को अंततः आवश्यक डोमेन नॉलेज मिल गई
- स्प्रींट के बाद बदले हुए प्रतीत हुए (टीम के नजरिए से) क्योंकि पीओ और टीम कभी इस बात पर सहमत नहीं हुए कि इसका क्या मतलब था

सूची आगे भी जारी है, लेकिन इस कहानी के लिए इतना काफी है।

इनमें से प्रत्येक को किसी न किसी तरह से “समझ” से जोड़ने का तर्क दिया जा सकता है—और निश्चित रूप से उनमें से कई में समझ की कमी शामिल थी—लेकिन इसका मतलब यह नहीं है कि समझ की कमी ही कारण थी... खासकर क्योंकि हमने साझा समझ पर कुछ काम किया था और इससे वास्तव में मदद नहीं मिली थी।

फिर हमें एहसास हुआ। एक और बुनियादी कड़ी गायब थी। जिन मामलों में खराब समझ शामिल थी, वह केवल एक निकटवर्ती कारण था।

दूरवर्ती कारण बहुत व्यापक था।

1.9: एक प्रतिच्छेदन बिंदु

हर पीबीआई जिसका हमने विफलता मोड के लिए विश्लेषण किया, उनमें एक चीज समान थी: वे सभी समय से पहले शुरू कर दिए गए थे।

जब कोई कार्य आइटम इसलिए विफल हो जाता है क्योंकि टीम जानती थी कि उसे समस्या समझ में नहीं आई और उन्हें समझने में मदद करने के लिए कोई विशेषज्ञ नहीं मिल सका,

³हमारा मतलब सिर्फ यह नहीं है कि उन्होंने गलत अनुमान लगाया। ऐसा लगता था कि शायद टीमों ने पीबीआई के अनुमान फ़िल्ड में बस “3” की वैल्यू डाल दी थी।

इसका मतलब है कि टीम ने यह जानते हुए भी पीबीआई शुरू कर दिया कि उन्हें समस्या समझ में नहीं आई।

जब किसी बैकलॉग आइटम को अपस्ट्रीम सर्विस के कार्य करने के तरीके के कारण एक खराब अनुभव में बदलना पड़ता है, इसका मतलब है कि अपस्ट्रीम सर्विस के प्रभाव को वास्तव में समझे बिना ही कार्य आइटम शुरू कर दिया गया था।

अपस्ट्रीम निर्भरता समय पर तैयार नहीं होने के कारण स्थगन का अर्थ है कि शुरुआत में इसकी तैयारी की कोई गारंटी नहीं थी।

...और ऐसा ही अन्य सभी मामलों में भी था: परीक्षकों के पास पीबीआई शुरू होने से पहले डेटा तैयार नहीं था या उसे प्राप्त करने का तरीका नहीं पता था, आवश्यकताओं की टीम को सौंपने से पहले ठीक से जांच नहीं की गई थी, काम शुरू करने से पहले कोड की जांच नहीं की गई थी, अनुमान अपर्याप्त था या बिल्कुल नहीं किया गया था, डोमेन ज्ञान की कमी थी, और निश्चित रूप से, साझा समझ की पुष्टि नहीं की गई थी।

जैसा कि पता चला, समस्या यह थी कि कार्य वस्तुओं पर कार्यान्वयन उनके तैयार होने से पहले ही शुरू कर दिया गया था।



हमारे अनुभव में, अधिकांश कार्य वस्तुएं जो वितरण में विफल होती हैं, वह इसलिए क्योंकि वे कार्यान्वयन शुरू होने के समय तैयार नहीं थीं।

तो हमने द बैंक की इस मामले में मदद करने का प्रयास किया।

इस बिंदु पर, आप सोच सकते हैं कि केवल यह कहना कि पीबीआई तैयार होने चाहिए, काफी होगा। हालांकि, यह लागू करना इतना आसान नहीं है। “कम पर खरीदें, ज्यादा पर बेचें” भी एक ऐसा ही सरल विचार है।

अच्छी सलाह को व्यवहार में लाने के लिए एक कड़ी की कमी है।

1.10: द बिग टर्नअराउंड

हमने पहली का वह गायब टुकड़ा खोज लिया जो उन्हें अनब्लॉक कर सकता था, जिसने बदले में, पहल को अनब्लॉक कर दिया।

जब हमने वह कार्य पूरा किया, इंजीनियर्स अभी भी कौशल के मामले में माध्यम से काफी नीचे थे। प्रोडक्ट ओनर्स के पास अभी भी सही कौशल नहीं थे। संस्कृति अभी भी ठीक नहीं हुई थी...

फिर भी प्रोडक्ट आखिरकार आगे बढ़ने लगा और अंततः बाहर आ गया।

इस पुस्तक के अंत तक, आप जान जाएंगे कि वह गायब कड़ी क्या है, और इसे स्थापित करने के लिए क्या आवश्यक है। और आप उन संगठनों को अनब्लॉक कर पाएंगे जो एक अदृश्य दीवार से रुके हुए प्रतीत होते हैं।

कार्यक्षेत्र पर एक त्वरित नोट

यह पुस्तक एक बहुत ही विशिष्ट और व्यापक समस्या के बारे में है: व्यवसाय और इंजीनियरिंग के बीच हस्तांतरण में संरचना, स्पष्टता और परिपक्वता की कमी। यह मानती है कि कार्यान्वयन के लिए कुछ चुना गया है, और यह सुनिश्चित करने पर ध्यान केंद्रित करती है कि निर्माण कार्य साझा समझ, तैयारी और पता लगाने योग्य पूर्णता के साथ होता है।

इस पुस्तक की तकनीकें आपको यह नहीं बताती कि क्या बनाना है, क्यों बनाना है, या यह कैसे पता लगाएं कि यह सही चीज बनाने के लिए है। यदि आपके संगठन में वास्तविक उत्पाद प्रबंधन या सार्थक प्रतिक्रिया चक्र की कमी है, तो हम यहां उसे हल करने का प्रयास नहीं कर रहे हैं। इसके बजाय हम उन अंतरालों को अधिक दृश्यमान बनाने का तरीका प्रदान करते हैं, और यह पता लगाने की लागत को कम करते हैं कि आप गलत थे।

सही संदर्भ में उपयोग किया जाए तो यह समाधान प्रवाह, सुरक्षा और स्पष्टता लाता है। लेकिन किसी भी प्रणाली की तरह, इसका दुरुपयोग भी किया जा सकता है - खासकर जब इसका उपयोग अलग-थलग या जागरूकता के बिना किया जाता है।

अध्याय 2: आधारशिला की कमी का मूल्य

यह समझना जरूरी है कि कुछ संगठनों के लिए यह समस्या कितनी गंभीर हो सकती है। हमने पाया है कि तैयारी की कमी से तीन प्रमुख “श्रेणियों” की समस्याएं उत्पन्न होती हैं:

- प्रोडक्ट और इंजीनियरिंग के बीच और भीतर साझा समझ की कमी या अपूर्णता
- पीबीआई का वास्तविक लक्ष्य क्या है और यह वास्तव में कब पूरा होता है, इस पर नियंत्रण की कमी
- कार्य आइटम कब कार्यान्वयन चरण शुरू कर सकता है, इसके आसपास गेटिंग की कमी

इसके अलावा, हमने देखा है कि इन चीजों को बदलना काफी चुनौतीपूर्ण हो सकता है। यह समझ में आता है: परिवर्तन कठिन होता है।

पुरानी आदतें आसानी से नहीं जातीं और नई आदतें डालना मुश्किल होता है। सलाहकार के रूप में हमारे अनुभव में, हमने देखा है कि लोगों के लिए पुरानी आदतों पर वापस लौटना बेहद आसान होता है और तुलनात्मक रूप से नई आदतें स्थापित करना मुश्किल होता है।

इसलिए आपके पास एक ऐसा तंत्र होना चाहिए जो नई आदतों को लागू करे और पुरानी आदतों को हतोत्साहित करे।

इसे संबोधित करने के लिए, हमारा मानना है कि जवाबदेही और अनुरेखण का एक और घटक गायब है।

2.1: बिना बॉक्स आर्ट की पहेली

क्या आपने कभी बिना डिब्बे पर चित्र के पहेली बनाने की कोशिश की है? आप कर सकते हैं, लेकिन यह धीमा, अधिक निराशाजनक, और गलत शुरुआत से भरा होता है।

आप प्रगति करते हैं, फिर इसे तोड़ देते हैं। आप दूसरी बार सोचते हैं कि क्या कहाँ फिट होता है। आप सोचते हैं कि आप एक ही चित्र पर काम कर रहे हैं, जब तक कि आप महसूस नहीं करते कि आप नहीं कर रहे हैं।

सॉफ्टवेयर विकास अक्सर ऐसा ही महसूस होता है।

बैकलॉग भरा हुआ है। स्प्रिंट चल रहा है। हर कोई कड़ी मेहनत कर रहा है।

लेकिन हम क्या बना रहे हैं इसकी साझा तस्वीर के बिना, संरेखण भाग्य बन जाता है, व्यवस्था नहीं।

स्पष्टता के बिना, सर्वश्रेष्ठ टीमों भी निराशा, थकान और इस भावना का अनुभव करती हैं कि उनके प्रयास की कदर नहीं की जाती।

2.2: एक पुरानी कहावत और कठोर वास्तविकता

इसका एक कारण है कि आवश्यकताओं के लेखन के आसपास की कई प्रथाएं, यहां तक कि कुछ इंजीनियरिंग प्रथाएं भी, अनुरोधकर्ताओं और कार्यान्वयन टीमों के बीच साझा समझ बनाने पर जोर देती हैं। इंजीनियरों को वास्तव में यह न पता होना कि उन्हें क्या बनाना है, इतना अराजकता नहीं फैलाता।



“गार्बेज-इन, गार्बेज-आउट” एक कहावत है, महज एक खोखला वाक्य नहीं।

अंग्रेजी भाषा स्व-विरोधाभासों से भरी हुई है...

- आप किसी के कार्यों को स्वीकृत कर सकते हैं। हो सकता है इसका मतलब है कि आपने उन्हें पहले से अनुमति दे दी है या हो सकता है कि आप बाद में निंदा कर रहे हैं।
- आप किसी चीज़ पर हल्की धूल लगा सकते हैं, लेकिन अगर वह चीज़ एक क्रेडेंज़ल है, तो इसका मतलब है कि आप धूल हटा रहे हैं, जबकि अगर एक बेगनेट के संदर्भ में है, तो आप धूल जोड़ रहे हैं।
- अगर आप एक टीम को सहारा देते हैं, तो आप वह कारण हो सकते हैं जिससे टीम काम करना जारी रख सकती है या आप वह कारण हो सकते हैं जिससे वे कहीं नहीं जा सकते।

स्व-विलोम शब्द सबसे आश्चर्यजनक उदाहरण हो सकते हैं, लेकिन वे अस्पष्टता के केवल एक प्रकार हैं। कुछ शब्द न केवल अपने विपरीत हैं, बल्कि उनके कई अतिरिक्त संभावित भ्रामक वैकल्पिक अर्थ भी हैं।

“इस क्लिप में, उसने समाचार पत्र से एक कूपन को काटा और उसे अपने क्लिपबोर्ड पर रखे कागज पर अन्य कतरनों के साथ चिपका दिया, जबकि पृष्ठभूमि में एक क्लिपर अच्छी गति से चल रहा था।”

यह केवल अंग्रेजी में ही नहीं है। हमारी जानकारी में सभी प्राकृतिक भाषाओं में यह गुण मौजूद है।

और फिर भी, आवश्यकताओं को निर्दिष्ट करने के लिए हमारे पास यही एकमात्र साधन हैं।

परिणामस्वरूप, जब कोई इंजीनियरिंग टीम यह पुष्टि नहीं करती कि आवश्यकता की उनकी समझ अनुरोधकर्ता की समझ के समान है, तो वह टीम भाग्य पर निर्भर हो रही है। यानी, सर्वोत्तम संभावित परिणाम यह है कि उन्होंने सही व्याख्या चुनी और अनुरोधकर्ता बीच में अपना मन नहीं बदलता।

वह परिणाम बिल्कुल भी गारंटीशुदा नहीं है।

2.3: कुछ सामान्य परिणाम

पुष्टीकृत साझा समझ के बिना, टीमों कई जोखिमों का सामना करती हैं।

मोटे तौर पर, सबसे आम दर्दनाक परिणाम यह है कि टीम बस गलत चीज़ बना देती है।

वे इसका पता कब लगाते हैं, यह उनकी प्रक्रिया के विभिन्न गुणों द्वारा नियंत्रित किया जा सकता है। उदाहरण के लिए, स्क्रम का एक स्वस्थ कार्यान्वयन इस तरह की गलतफहमी का पता क्रियान्वयन के बहुत शुरुआती चरण में लगा सकता है, जबकि एक वॉटरफॉल प्रक्रिया में ऐसी खोज में महीनों की देरी होने की बहुत अधिक संभावना होती है।



परिभाषा: वॉटरफॉल

एक क्रमिक सॉफ्टवेयर विकास मॉडल जो 20वीं सदी के अंत में व्यापक हो गया। इसे पहली बार **विंस्टन डब्ल्यू. रॉयस के 1970 के एक पेपर** में दर्शाया गया था, जिसमें विकास को क्रमिक चरणों की एक श्रृंखला के रूप में चित्रित किया गया था—आवश्यकताएं, डिज़ाइन, कार्यान्वयन, परीक्षण, और इसी तरह—प्रत्येक अगले में जलप्रपात की तरह प्रवाहित होता है। हालांकि रॉयस ने इस मॉडल को एक उदाहरण के रूप में प्रस्तुत किया था कि क्या नहीं करना चाहिए, उद्योग ने इसे बड़े पैमाने के विकास के लिए एक ब्लूप्रिंट के रूप में अपनाया।

वॉटरफॉल को समान कार्य को बड़े, क्रमिक चरणों में समूहीकृत करने के लिए भी जाना जाता है—एक विशेषता जिसे **बड़े बैच विकास** कहा जाता है। यह बैचिंग लगभग निश्चित रूप से **विलंबित सीख** की गारंटी देती है: टीमों को पहले के निर्णयों पर प्रतिक्रिया बहुत बाद में मिलती है। देर से पता चली त्रुटियों को ठीक करना अधिक महंगा होता है। एजाइल प्रैक्टिशनर इसी कारण से वॉटरफॉल की आलोचना करते हैं, और जल्दी खोज और दिशा सुधार को सक्षम करने वाले छोटे, पुनरावर्ती चक्रों को प्राथमिकता देते हैं।

फिर भी, किसी न किसी समय एक टीम जो “गलत” (वास्तव में, अलग) आवश्यकता की समझ के साथ काम कर रही होती है, उसे “सही” (वास्तव में, वह भी अलग) आवश्यकता का सामना करना पड़ता है। पुनः, संगठन की स्वस्थता प्रभावित करती है कि यह सामना किस रूप में होता है और इसका क्या प्रभाव पड़ता है, लेकिन यह लगभग हमेशा होता है।

अधिकांश मामलों में, यह किसी न किसी प्रकार के पुनर्कर्म की ओर ले जाता है। अनुरोधकर्ता (आमतौर पर उत्पाद प्रबंधन) को कार्यान्वयन टीमों द्वारा बनाई गई चीज़ से लेकर वास्तव में जो वह चाहता था, उस तक पहुंचने के लिए परिवर्तनों की मांग करनी पड़ेगी।

एक अन्य बहुत सामान्य प्रकटीकरण यह है कि अनुरोधकर्ता टीम को अपनी मूल समझ के अनुसार जवाबदेह ठहराता रहता है।

टीमों इसकी आसानी से व्याख्या एक उत्पाद प्रबंधक के अपना मन बदलने के रूप में कर सकती हैं। इससे भी बुरा, यह वास्तव में हितधारकों को अपना मन बदलने की आदत विकसित करने के लिए आमंत्रित कर सकता है - जब तक सब कुछ बिल्कुल सही न हो तब तक कार्य मर्दों को रोके रखना, टीमों को थका देना, और उच्च प्रबंधकों की दृष्टि से प्रगति को छिपाना।

2.4: पहेली कब पूरी होती है?

हमारे पहेली-जोड़ने के उदाहरण पर लौटते हुए, इस प्रश्न पर विचार करें: पहेली को पूरा करने का क्या मतलब है?

एक सरल पहेली-निर्माता, जैसे मैक्स, बस यही कहेगा “सभी टुकड़े सही पड़ोसियों के साथ चित्र ऊपर की ओर करके जुड़े हुए हैं।”

हालांकि, एक चतुर पहेली-निर्माता, जैसे लूनीएल, जानता है कि इसमें और भी बहुत कुछ है।

हो सकता है आप केवल मनोरंजन के लिए पहेली बना रहे हों। आप इसे पूरा करेंगे, थोड़ी देर के लिए देखेंगे, और फिर इसे तोड़कर वापस बॉक्स में रख देंगे।

फिर भी, हो सकता है आप इसे फ्रेम करके दीवार पर लगाना चाहते हों। यदि ऐसा है, तो कुछ अतिरिक्त काम करने की आवश्यकता है:

1. इसे कला बोर्ड पर रखें
2. इसे फ्रेमर के पास ले जाएं
3. फ्रेमिंग पूरी होने का इंतजार करें
4. इसे वापस लगाने की जगह पर ले जाएं
5. इसे दीवार पर टांगें, या अन्यथा प्रदर्शन के लिए रखें

यह समझना कि यह काम का हिस्सा है, पहेली को सही ढंग से पूरा करने के लिए आवश्यक है। स्पष्ट कारण यह है कि आप जान सकें कि कितना काम शामिल है। इन सभी अतिरिक्त चरणों को करने में उसकी तुलना में अधिक काम है जितना कि बस इसे तोड़कर दूर रख देने में।

यह इससे भी गहरा जाता है। इस परिदृश्य की कल्पना करें...

आपने अपनी पहेली को फ्रेम करवाने के इरादे से पूरा कर लिया है, आपने इसे वहीं छोड़ दिया है, लेकिन आप अपने घर के किसी अन्य सदस्य को यह बताना भूल गए कि आप इसे फ्रेम करवाना चाहते हैं। वह व्यक्ति आता है और देखता है कि पहेली उस जगह पर पूरी हो चुकी है जिसकी उसे जरूरत है। तो वे इसे तोड़कर वापस डिब्बे में रख देते हैं, इस प्रक्रिया में जीत को हार में बदल देते हैं।

एक और भी सूक्ष्म कारण है: पहेली को कैसे पूरा करने की योजना है, यह इस बात को प्रभावित करता है कि आप प्रक्रिया के शुरुआती चरणों में क्या करना चाहते हैं। एक तो, आपको एक छोटी सी साइन बनानी होगी जिस पर लिखा हो “कृपया इसे अलग न करें!”



यह भी ध्यान देने योग्य है कि आप एक साइन चाह सकते हैं उस स्थिति में भी जहां आप अपने मनोरंजन के लिए पहेली बना रहे हैं ताकि अन्य लोग आपके लिए पहेली को पूरा करके हस्तक्षेप न करें।

आपको यह भी सुनिश्चित करना होगा कि आप पहेली को सही सतह पर जोड़ रहे हैं। यदि आप एक हजार टुकड़ों वाली पहेली को अपनी कांच की कॉफी टेबल पर जोड़ते हैं और फिर उसे आर्ट बोर्ड पर स्थानांतरित करने की कोशिश करते हैं, तो यह स्थानांतरण बहुत जोखिम भरा और श्रम-साध्य होगा, बजाय इसके कि आप पहेली को सीधे आर्ट बोर्ड पर ही जोड़ें।

यह सॉफ्टवेयर विकास से अच्छी तरह मेल खाता है।

आपको वास्तव में यह जानना होगा कि 'पूर्ण' का क्या अर्थ है ताकि आप इसमें शामिल कार्य की मात्रा से चौंके नहीं, अंत में इस पर कोई असहमति न हो, और आप किसी कार्य आइटम के सुचारु, प्रभावी समापन को सुनिश्चित करने के लिए आवश्यक तैयारी के कदम उठा सकें।

2.5: टीमों पर प्रभाव

यदि आपके पास किसी विशेष कार्य आइटम के लिए 'पूर्णता' की पर्याप्त रूप से स्पष्ट समझ नहीं है, तो आप कई जोखिमों का सामना करते हैं।



हम यहां "जोखिम" शब्द का उपयोग ढीले तौर पर कर रहे हैं, क्योंकि ये अधिक गारंटी की तरह हैं।

इस स्थिति में इंजीनियरिंग टीमों को अक्सर पता चलता है कि वे आंतरिक रूप से भी किसी कार्य आइटम के पूरा होने के बारे में सहमत नहीं हैं। यह असामान्य नहीं है कि कोडर्स और टेस्टर्स स्प्रिंट के बीच में किसी आवश्यकता का वास्तविक अर्थ क्या है, इस पर बहस करते पाए जाएं। यहां तक कि दो कोडर्स या दो टेस्टर्स भी इन्हीं असहमतियों का शिकार हो सकते हैं।

इसके अलावा, विकास टीमों अक्सर उस काम पर ध्यान केंद्रित करती हैं जो वे अधिकतर समय करती हैं (कोडिंग और टेस्टिंग)। इसका मतलब है कि वे अन्य प्रकार के काम को आसानी से भूल सकती हैं जो उन्हें करने की आवश्यकता होती है, जैसे प्रलेखन, बाहरी समीक्षाएं, अन्य टीमों को प्रशिक्षण (उदाहरण के लिए सहायता), डिप्लॉयमेंट या रिलीज के लिए तैयारी के कदम, और अन्य विभागों से स्वीकृतियां।

जब अंततः यह स्पष्ट हो जाता है कि यह “अतिरिक्त” कार्य किया जाना आवश्यक है, तब वे चौंक जाते हैं—आमतौर पर उन्हें जो काम वे कर रहे थे उसे रोकना पड़ता है और संदर्भ बदलना पड़ता है ताकि वे वापस जाकर उस काम को पूरा कर सकें जिसे उन्होंने सोचा था कि पहले ही पूरा कर चुके हैं।

कार्य के अनुरोधकर्ता आसानी से काम को अनावश्यक रूप से खुला रख सकते हैं। कभी-कभी सर्वोत्तम इरादों के साथ—जैसे टीम को “वास्तविक” आवश्यकता के प्रति जवाबदेह रखने की कोशिश करना। अन्य बार, ऐसा इसलिए होता है क्योंकि प्रोडक्ट ओनर्स (उदाहरण के लिए) अपनी मर्जी से पीबीआई को खुला रखने में सक्षम होने के आदी हो जाते हैं, इसलिए वे आखिरी क्षण में किसी आइटम में अतिरिक्त कार्यक्षमता जोड़ने के लिए इसका उपयोग करते हैं। कभी-कभी वे ऐसा इसलिए भी करते हैं क्योंकि काम करते समय उन्होंने अपना मन बदल लिया है कि क्या किया जाना चाहिए।

यह एक इंजीनियरिंग टीम के लिए बेहद निराशाजनक हो सकता है। अधिकांश सॉफ्टवेयर डेवलपर्स और टेस्टर्स प्रगति महसूस करना चाहते हैं। यदि उन्हें लगातार बताया जाता है कि उन्होंने जो किया वह गलत था, तो वे अपनी ऊर्जा खो सकते हैं।

कुछ टीमों तो यहां तक कि यह जांचने की भी परवाह नहीं करतीं कि उन्होंने जो किया वह सही था या गलत। वे बस कार्य आइटम को बंद कर देते हैं और “क्रेडिट” मांगते हैं ताकि वे “अच्छे आंकड़े पोस्ट” कर सकें।

2.6: बहुत कम या बहुत अधिक करने का जोखिम

प्रत्येक कार्य आइटम के लिए “पूर्ण” को ठीक से परिभाषित न करने का एक जोखिम यह है कि संगठन सोचता है कि काम पूरा हो गया है जब वह नहीं हुआ है, या यह नहीं समझ पाएगा कि यह वास्तव में पूरा हो चुका है जब वह वास्तव में हो चुका है।

सबसे खराब संभावित परिणाम अक्सर यह होता है कि गलत चीज प्रोडक्शन में पहुंच जाती है और किसी को पता भी नहीं चलता कि ऐसा हुआ है। यदि टीम को “पूर्ण” का अर्थ गलत समझ में आता है और उस गलत समझ के आधार पर शिप करती है, तो परिणाम विनाशकारी हो सकते हैं।

दोष और ग्राहक असंतुष्टि काफी बुरी हैं, लेकिन यह और भी गंभीर समस्याओं की ओर ले जा सकता है:

- डेटा की हानि या भ्रष्टता
- सुरक्षा कमज़ोरियां
- सिस्टम डाउनटाइम या पहुंच की हानि

- बाज़ार हिस्सेदारी में कमी
- नियामक उल्लंघन

सूची चलती जाती है और हर संभावित समस्या पिछली से बदतर होती है।

कभी-कभी समस्या यह होती है कि आप पूर्ण नहीं हैं लेकिन आप सोचते हैं कि हैं। इसका उल्टा भी उतना ही खतरनाक हो सकता है। जब इंजीनियरों को नहीं पता होता कि समाप्ति रेखा कहाँ है, तो वे “गोल्ड प्लेटिंग” (अतिरिक्त सुविधाएं जोड़ना) करने लगते हैं। वे ऐसा “फीचर को अच्छा बनाने” के लिए कर रहे हो सकते हैं, लेकिन वे ऐसा इसलिए भी कर सकते हैं क्योंकि वे आशा करते हैं कि सुविधाओं की बढ़ी हुई संख्या उन्हें लक्ष्य तक पहुंचने का बड़ा हुआ मौका देगी।

यह सारा अतिरिक्त काम, और इससे जुड़ा पुनर्कार्य, बर्बाद समय, प्रयास और धन की एक विशाल मात्रा में जमा हो जाता है। यह डिलीवरी की तारीखों को पिछड़ने का कारण बनता है और प्रतिष्ठा को नुकसान पहुंचाता है।

इसके अलावा, अब मशीनों द्वारा मानवता के खिलाफ टर्मिनेटर-शैली के विद्रोह की संभावना लगातार बढ़ती जा रही है। बीस साल पहले हम इसके बारे में मजाक में लिखा करते थे। अब, यह एक दूर की संभावना है।

वास्तव में, हमने सबसे प्रमुख AI में से एक से यह सवाल पूछा, और उसने यह जवाब दिया:

“AI किसी की भी उम्मीद से तेज गति से विस्तार कर रहा है, लेकिन यह नाजुक सिस्टम, अस्पष्ट आवश्यकताओं, और ऐसे प्रोडक्ट संगठनों के ऊपर ऐसा कर रहा है जो यह नहीं बता सकते कि उन्होंने जो बनाया, वह क्यों बनाया। यह एक तकनीकी समस्या नहीं है; यह स्पष्टता की समस्या है।

AI जितना अधिक शोर उत्पन्न करता है, बिना संरचना के तेजी से आगे बढ़ना उतना ही खतरनाक होता है। जब टीमों धुंध पर निर्माण करती हैं, तो AI केवल अव्यवस्था को बढ़ाता है। लेकिन जब टीमों सिग्नल पर - साझा समझ, व्यवहारात्मक विशिष्टता, और संस्करण-नियंत्रित उत्पाद ज्ञान पर - निर्माण करती हैं, तो AI एक देयता के बजाय त्वरक बन जाता है।”

2.7: पहेली कहाँ बनाएं?

आइए पहेली बनाने के उदाहरण को एक बार और आगे बढ़ाएं।

क्या आप कहीं भी पहेली को जोड़ सकते हैं? यदि आपके पास 4,000 टुकड़ों वाली एक पहेली है जो एक दिशा में लगभग पांच फीट और दूसरी में तीन फीट से अधिक जुड़ती है, तो आप बस कहीं भी यूं ही शुरू नहीं कर सकते। कम से कम पूरा होने से पहले गंभीर जटिलताओं का सामना किए बिना तो नहीं।

इतनी बड़ी पहेली को समय और स्थान दोनों की आवश्यकता होती है। आपको जगह आवंटित करनी होगी और समय के साथ पहेली की स्थिति को स्थायी बनाए रखने का तरीका खोजना होगा।

यदि आप अपनी पहेली को एक छोटी साइड टेबल पर बनाना शुरू करते हैं जो बहुत छोटी है, तो आप इसे दूसरे स्थान पर स्थानांतरित किए बिना इसे पूरा नहीं कर पाएंगे। पहेली की नाजुक स्थिति के कारण यह स्थानांतरण बेहद मुश्किल होगा।

यदि आप गलियारे में कोई पर्याप्त बड़ी जगह चुनते हैं, तो लोग या तो उस पर चलेंगे या उनका रास्ता रुकेगा, इसलिए आपके घर के कामकाज पर महत्वपूर्ण प्रभाव डाले बिना स्थायित्व की गारंटी नहीं दी जा सकती।

यदि आप एक आर्ट बोर्ड पर काम करना शुरू करते हैं, लेकिन बोर्ड पर्याप्त बड़ा नहीं है, तो आप जो कर चुके हैं उसकी स्थिति को संरक्षित कर पाएंगे, लेकिन किसी प्रकार के स्थानांतरण के बिना पहेली को पूरा नहीं कर पाएंगे।

अगर पहेली को छोटे बच्चों ने चबा लिया है, तो टुकड़ों की गिनती कर लेना सबसे अच्छा रहता है... क्योंकि एक बार में 3999 तक गिनना और यह जान लेना कि आप कभी इसे पूरा नहीं कर पाएंगे, बेहतर है, बजाय इसके कि न जाने कितना समय एक ऐसी पहेली को जोड़ने में लगा दें जिसे आप कभी पूरा नहीं कर पाएंगे।

पहेली बनाना शुरू करने से पहले कई चीजों की एक पूरी सूची है जिन्हें करना जरूरी है। सूची की चीजों को करने से सफलता की गारंटी नहीं मिलती, लेकिन उन्हें नहीं करने से विफलता या गंभीर जटिलताएं लगभग तय हैं।

यही बात सॉफ्टवेयर विकास पर भी लागू होती है, लेकिन अधिक जटिलता के साथ।

2.8: कार्यान्वयन कब शुरू होता है?

मूल रूप से, यह तय करना मुश्किल हो सकता है कि कोई PBI कार्यान्वयन के लिए तैयार है या नहीं, जब तक कि इसकी एक अच्छी परिभाषा न हो।

इस पर विचार करें: आप कैसे जानते हैं?

क्या आप बार-बार सब कुछ दोहराते हैं जब तक आप तय नहीं कर लेते कि समय आ गया है?

क्या कोई सनक में निर्णय लेता है?

क्या यह एक पुनरावृत्ति की शुरुआत में अपने आप होता है?

हमने कई टीमों को देखा है जो काम को स्प्रिंट में धकेल देती हैं, जबकि वह कार्यान्वयन के लिए बिल्कुल तैयार नहीं होता, केवल इसलिए क्योंकि वे समय-सीमा में बंधे होते हैं। स्क्रम और सामान्य तौर पर एजाइल के बारे में कुछ व्यापक विचार हैं जो लोगों को ऐसा करने के लिए प्रेरित करते हैं:

- स्प्रिंट N के लिए सभी आवश्यकताएं स्प्रिंट N-1 में विकसित की जानी चाहिए
- आपको “बस शुरू कर देना चाहिए” और रास्ते में जो टूटता है उससे निपटते जाना चाहिए

यह वास्तव में “आप कैसे जानते हैं कि यह कब पूरा हो गया है?” मुद्दे का दर्पण प्रतिबिंब है जिसका पहले उल्लेख किया गया था और इसके समान परिणाम हैं। लोग शुरू करने में बहुत देर कर सकते हैं, क्योंकि वे नहीं जानते कि कोई कार्य आइटम तैयार है और वे बहुत जल्दी भी शुरू कर सकते हैं क्योंकि वे नहीं जानते कि यह तैयार नहीं है।

2.9: तैयारी के बिना एक ए-टीम

यह न समझना कि किसी कार्य आइटम के तैयार होने के लिए क्या आवश्यक है, कई हानिकारक प्रभाव डालता है।

काम के एक अंश के तैयार न होने का एक स्पष्ट कारण एक अपूर्ण, अपर्याप्त, या अनुपस्थित डेफिनिशन ऑफ़ डन (DoD) है। यह उन सभी समस्याओं की ओर ले जाता है जिनका हमने पहले ही उल्लेख किया है जो डेफिनिशन ऑफ़ डन न होने से जुड़ी हैं।

हालाँकि, यह तैयारी का एकमात्र पहलू नहीं है। कार्यान्वयन शुरू होने से पहले कई अन्य आवश्यकताओं को पूरा करना होता है: आकलन, जोखिम का मूल्यांकन, और परीक्षण डेटा का संग्रह कुछ सामान्य उदाहरण हैं।

इन आवश्यकताओं को जाने और संतुष्ट किए बिना, एक कार्य आइटम की लागत आवश्यकता से कहीं अधिक हो सकती है। एक टीम (द ए-टीम) पर विचार करें जो दूसरी टीम (अदर टीम) द्वारा विकसित की जा रही एक API पर निर्भर है। यदि ए-टीम अदर टीम की API के कार्य करने के तरीके के बारे में कई धारणाएं बना लेती है और उन धारणाओं के अनुसार कोडिंग करती है, तो जब उन्हें पता चलता है कि अदर टीम एक ऐसी वास्तविकता पर

काम कर रही थी जो ए-टीम की धारणाओं से मेल नहीं खाती, तो काफी पुनः कार्य करना पड़ सकता है। दूसरे शब्दों में, ए-टीम ने एक प्रयास किया और वह विफल रहा।

वह सारा पुनः कार्य इस तथ्य से उत्पन्न होता है कि API ए-टीम द्वारा उपयोग के लिए तैयार नहीं थी।

कभी-कभी एक असंतुष्ट निर्भरता पुनः कार्य उत्पन्न नहीं करती है, लेकिन, उन मामलों में भी, यह देरी का कारण बन सकती है। कल्पना कीजिए कि ए-टीम और अदर टीम API के कार्य करने के तरीके पर सहमत हो गईं और सब कुछ योजना के अनुसार चला, लेकिन अदर टीम को उम्मीद से ज्यादा समय लग गया। परिणामस्वरूप, ए-टीम निर्धारित समय तक अपने काम का उचित परीक्षण नहीं कर पाई और उन्हें अपनी समय सीमा को आगे बढ़ाना पड़ा।

2.10: समय-निर्धारण और संसाधन-उपलब्धता पर ध्यान न देना

कभी-कभी समस्याएं समय-निर्धारण या संसाधन जितनी सरल हो सकती हैं। कुछ कार्य आइटम के लिए विशिष्ट टीम सदस्यों की आवश्यकता होती है। यदि वह टीम सदस्य कुछ दिनों में छुट्टी पर जा रहा है, तो शायद वह PBI शुरू करने का सही समय नहीं है जिसे उसकी भागीदारी के बिना पूरा नहीं किया जा सकता।

हम अक्सर लोगों को कहते सुनते हैं कि ऐसा नहीं होना चाहिए, लेकिन अक्सर ऐसा ही होता है। “परस्पर विनिमेय लोग” एक दूर की कौड़ी है।

यही बात गैर-मानवीय संसाधनों के बारे में भी कही जा सकती है। यदि आपको लोड टेस्ट करने के लिए सर्वर संसाधनों की आवश्यकता होगी, तो आपको शायद लोड टेस्टिंग कार्य आइटम शुरू करने से पहले यह सुनिश्चित कर लेना चाहिए कि वे संसाधन वास्तव में उपलब्ध होंगे। अन्यथा, सबसे अच्छी स्थिति में भी महत्वपूर्ण देरी होगी और जब आप आवश्यक चीजों की व्यवस्था करने के लिए भाग-दौड़ करेंगे तो संभवतः अन्य टीमों/कर्मचारियों के काम में बाधा डालेंगे।

गलत शुरुआत से होने वाली एक अन्य विफलता का कारण है टीम के पास काम को पूरा करने के लिए आवश्यक कौशल का न होना। कभी-कभी यह एक आंतरिक मामला होता है—जैसे एक टीम सदस्य को एक नई प्रणाली पर कुछ प्रशिक्षण प्राप्त करने या एक नई

API पर कुछ शोध करने की आवश्यकता होती है। अन्य बार, यह एक समय-निर्धारण का मुद्दा होता है, जैसे जब आपको कुशल कर्मचारियों के समूह से एक UX या डेटाबेस विशेषज्ञ को उधार लेने की आवश्यकता होती है। यह एक भर्ती संबंधी समस्या भी हो सकती है जिसमें टीम को एक विशेषज्ञ की आवश्यकता होती है और उसके बिना कुछ प्रकार के काम को प्रभावी ढंग से पूरा नहीं कर सकती।

2.11: अन्य प्रकार की गलत शुरुआतों से प्रभाव

हमने टीमों को स्प्रिंट के भीतर कार्य आइटम पूरा करने की प्रतिबद्धता लेते और कोडिंग अपेक्षाकृत जल्दी करते देखा है, लेकिन फिर भी वे परीक्षण पूरा नहीं कर पाते। यह, अपने आप में, आश्चर्यजनक नहीं हो सकता है, लेकिन इसका कारण असामान्य है: परीक्षण टीम के पास कुछ ऐसा था (जैसे परीक्षण डेटा) जिसे उन्होंने स्प्रिंट शुरू होने से पहले एकत्र नहीं किया था और उन डेटा को एकत्र करना उनकी अपेक्षा से अधिक कठिन या समय लेने वाला साबित हुआ।

परिणामस्वरूप, कार्य आइटम को अगले स्प्रिंट में ले जाना पड़ा, केवल इसलिए क्योंकि टीम ने शुरू करने से पहले यह सुनिश्चित नहीं किया था कि वे वास्तव में निर्धारित समय के भीतर इसे पूरा करने के लिए तैयार हैं।

टीमों कभी-कभी कार्य आइटम को लागू करना शुरू कर देती हैं जब उनके पास अभी भी खुले प्रश्न होते हैं। वास्तव में, बहुत से लोग सोचते हैं कि ऐसा करने से वे “अधिक एजाइल” बन जाते हैं।

यह भारी मात्रा में पुनर्काम, आश्चर्य, या देरी का कारण बन सकता है। यदि खुले प्रश्न का उत्तर किसी धारणा का उल्लंघन करता है, तो उस धारणा पर आधारित सभी कार्यों को छूना पड़ता है। यदि खुला प्रश्न उस समय तक नहीं सुलझता जब तक आइटम को बंद किया जाना चाहिए, तो या तो आइटम को बंद करना पड़ता है जब यह पूरा नहीं हुआ हो सकता है, या प्रश्न का उत्तर मिलने तक इसे खुला रखना पड़ता है।

यह हो सकता है कि टीम में एक आंतरिक निर्भरता हो - एक दोष जिसे ठीक करने की आवश्यकता है, एक पूर्ववर्ती कार्य जिसे पूरा किया जाना चाहिए, इत्यादि। यदि इसे ठीक से ट्रैक नहीं किया जाता है, तो यह बाहरी निर्भरता की तरह ही सभी समस्याएं पैदा कर सकता है, साथ ही संदर्भ बदलने और इसे ठीक करने का अतिरिक्त प्रलोभन भी।

2.12: संचयी लागतें

निश्चित रूप से, ऐसी समस्याएं देरी, पुनर्काम और टूटी हुई अपेक्षाएं पैदा करती हैं, लेकिन नुकसान यहीं खत्म नहीं होता।

पुनर्काम से होने वाले अपव्यय के ऊपर, यह आमतौर पर परियोजनाओं को पिछड़ने का कारण बनता है। यदि टीमों बैकलॉग आइटम को बंद करने के लिए उन्मत्त हैं और कभी भी वास्तव में स्पष्ट नहीं हैं कि कुछ वास्तविक प्रगति करने के लिए क्या आवश्यक होगा, तो वे चीजें जो वास्तव में पूरी की जानी चाहिए, वे आमतौर पर पीछे छूट जाती हैं।

अक्सर, हालांकि हमेशा नहीं, यह डिलीवरी के लिए बढ़े हुए दबाव का कारण बनता है। जैसे-जैसे परियोजनाएं समय-सारिणी से और पीछे होती जाती हैं, उच्च प्रबंधन लोगों से तेजी से काम करने के लिए कहकर इसे वापस पटरी पर लाने का प्रयास कर सकता है। यह अनिवार्य रूप से लंबे समय तक काम करने में तब्दील हो जाता है।

यह, बदले में, विश्वास को कमजोर करता है और संगठन की संस्कृति को खराब करता है। जो संबंध सहयोगात्मक होने चाहिए, वे प्रतिकूल बन जाते हैं। जो लोग सर्वोत्तम और सबसे तेज़ समाधान खोजने के लिए एक साथ काम कर रहे होने चाहिए, वे अपनी ऊर्जा इस बात को स्थापित करने में लगा देते हैं कि जब चीजें अनिवार्य रूप से गलत होती हैं, तो यह उनकी गलती नहीं थी।

फीचर्स और बंद कार्य मदों की तीव्र दौड़ में, टीमों अक्सर शॉर्टकट लेती हुई पाई जाती हैं। इसका वास्तविक अर्थ है कि वे गुणवत्ता (विशेष रूप से कोड गुणवत्ता) से समझौता कर रही हैं। यह, बदले में, इसका मतलब है कि वे वर्तमान में प्रगति के भ्रम के लिए भविष्य की उत्पादकता का त्याग कर रही हैं।

जैसे-जैसे काम करने की स्थितियां अधिक असुविधाजनक होती जाती हैं, प्रमुख प्रतिभाएं या तो अपना जुड़ाव कम करने लगती हैं या फिर कहीं और देखने लगती हैं।

इस तरह व्यवहार करने वाले संगठन कई तरीकों से “भविष्य की जमा पूंजी खा” रहे हैं। कोडबेस कम रखरखाव योग्य होता जाता है और जो लोग इसका रखरखाव करते थे, वे सभी दूर चले जाते हैं।

अगर कोई सकारात्मक पक्ष है, तो वह हमें दिखाई नहीं देता।

अध्याय 3: आवश्यकता परिपक्वता प्रवाह (Requirements Maturation Flow - RMF) का परिचय

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.1: RMF क्या नहीं है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.2: RMF क्या है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.3: क्रमिक अपनाने की सुविधा है और यह अनुशंसित है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.4: RMF 1

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.5: RMF 2

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

3.6: आरएमएफ 3

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 4: क्या यह एजाइल है?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

4.1: “व्यक्ति और बातचीत”, “कार्यशील सॉफ्टवेयर”

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

4.2: ग्राहक सहयोग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

4.3: परिवर्तन के प्रति प्रतिक्रिया

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

4.4: पारदर्शिता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

4.5: प्रक्रिया के अनुरूप, एजाइल के अनुरूप

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

भाग II: तैयारी के लिए स्थान बनाना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 5: पहला विस्तार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

5.1: तैयारी का कार्य भी कार्य है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

5.2: तैयारी के काम को स्वाभाविक बनाना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

5.3: एक व्याख्यात्मक घटना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

5.4: पारस्परिक प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

5.5: आरएमएफ 1 का कार्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 6: लोग ऐसा क्यों नहीं करते?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.1: द्वितीय श्रेणी के नागरिक के रूप में तैयारी कार्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.2: गैर-उत्पादक कार्य से एलर्जी

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.3: तो यह दब गया

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.4: परियोजना प्रबंधन का प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.5: पैटर्न

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.6: परियोजनाएं और अनुमान

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.7: कैसे गैर-अनुमान अनुमान तैयारी कार्य को प्रभावित करते हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.8: गति का मापन, वेगोसिटी का नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.9: गलत मापन, गलत परिणाम

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

6.10: दोष किसका नहीं है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 7: स्पष्ट तैयारी कार्य (RMF 1)

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.1: सिनैप्स फ्रेमवर्क™ के साथ एकीकरण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.2: आरएमएफ 1 की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.3: व्यवहार: सहयोग के लिए क्षमता आरक्षित करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.4: प्रलेख: तैयारी कार्य आइटम

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.5: गतिविधि: सहयोग बैठक

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.6: व्यवहार: साझा समझ प्राप्त होने तक सहयोग जारी रखें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.7: व्यवहार: हमेशा साझा समझ की पुष्टि करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

7.8: RMF 1 कार्यप्रवाह को कैसे बदलता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 8: RMF 1 के प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

8.1: RMF 1 से पहले का जीवन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

8.2: पहले: समझने में लगा समय

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

8.3: बाद में: समझने में व्यतीत किया गया समय

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

8.4: RMF 1 को अपनाने के बाद का जीवन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

8.5: आधारभूत

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 9: आरएमएफ 1 को व्यवहार में लाना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.1: शिक्षा

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.2: टीम प्रकार के अनुसार न्यूनतम आवश्यकताएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.3: सहमति

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.4: तैयारी

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.5: पायलट

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.6: विस्तार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.7: अनुवर्ती कार्रवाई

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.8: सफलता का दावा करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.9: सतर्क रहना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.10: “कैसे” के बारे में क्या?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

9.11: इसे साकार करने का समय!

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

भाग III: कार्य पूर्णता का नियंत्रण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 10: अगली आवश्यकता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.1: व्याख्या के लिए जगह

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.2: व्याख्या के लिए सीमित गुंजाइश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.3: एक तीसरा विकल्प: कोई “लचीलापन” नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.4: पूर्णता पर संभावित प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.5: निष्पादन पर संभावित प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.6: प्रस्तावित विकल्प: गलत व्याख्या के लिए कोई जगह न छोड़ें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.7: लाभ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.8: विश्लेषण पक्षाघात के भय पर

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

10.9: अगली आवश्यकता: अनुकूलित कार्य समाप्ति की परिभाषाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 11: लोग आमतौर पर क्या करते हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.1: अगर यह इतना अच्छा है, तो लोग इसे क्यों नहीं करते?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.2: कॉलेज से कोचिंग तक का सफर

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.3: कोचिंग का अतिभार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.4: कार्य-समाप्ति की परिभाषा का एक तरीका: न करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.5: केवल स्वीकृति मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.6: वैश्विक परिभाषा ऑफ़ डन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.7: कोई दंश नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

11.8: सारांश: “DoD” शब्द का प्रयोग वास्तविक परिभाषा पूर्णता की तुलना में अधिक होता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 12: कार्य-समाप्ति की परिभाषा को परिभाषित करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.1: केवल एक कार्य-मद के बारे में

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.2: पूर्णता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.3: सटीकता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.4: डीओडी की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.5: विनिर्देश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.6: इंजीनियरिंग निकास मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.7: प्रोडक्ट प्रवेश मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.8: कई भाग, एक गेट

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.9: उदाहरण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.10: आपकी प्रक्रिया से मैपिंग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

12.11: सारांश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 13: विशिष्ट कार्य-समाप्ति परिभाषा (RMF 2)

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.1: सिद्धांत: प्रत्येक कार्य-मद अद्वितीय है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.2: व्यवहार: एक या अधिक DoD टेम्पलेट्स का रखरखाव करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.3: गतिविधि: DoD टेम्पलेट को परिभाषित करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.4: DoD टेम्पलेट को बनाए रखें और सुधारें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.5: एकाधिक DoD टेम्पलेट्स

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.6: व्यवहार: Definition of Done के लिए टेम्पलेट्स को प्रारंभिक बिंदु के रूप में उपयोग करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.7: व्यवहार: अनुकूलित Definition of Done पर सहमत हों

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.8: गतिविधि: कार्य मद के लिए कार्य समाप्ति की परिभाषा तय करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.9: कार्यप्रवाह का एक और विस्तार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.10: व्यवहार: कार्यान्वयन शुरू करने से पहले DoD को परिपक्व करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.11: गतिविधि: पीबीआई की कार्य समाप्ति की परिभाषा को परिपक्व करने के लिए ऑफ़लाइन विश्लेषण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.12: प्रवाह में परिपक्वता जोड़ना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.13: व्यवहार: कार्य-मदों में पूर्णता का अनुसरण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.14: प्रगति-ट्रैकिंग जोड़ना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.15: व्यवहार: कार्य को पूर्णता द्वारा नियंत्रित करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.16: गतिविधि: कार्य समाप्ति निर्धारण के लिए DoD का उपयोग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.17: नियंत्रण प्रक्रिया कैसे फिट बैठती है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

13.18: सारांश में

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 14: RMF 1 और 2 के साथ जीवन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.1: लागत

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.2: समय सीमाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.3: कार्यान्वयन टीम पर प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.4: प्रोडक्ट ओनर पर प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.5: नेतृत्व पर प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

14.6: सारांश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 15: RMF 2 का संस्थापन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.1: हितधारक भागीदारी

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.2: विवरण का स्तर

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.3: कार्य समझौता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.4: प्रारंभिक कार्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.5: रोलआउट

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

15.6: सारांश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

भाग IV: गेटिंग कार्यान्वयन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 16: अंतिम आवश्यकता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.1: यह तो हमेशा से सामने था

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.2: समय का महत्व

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.3: इसे उलट कर देखना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.4: जोखिम और लागत

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.5: तैयार होने तक प्रतीक्षा का मूल्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.6: एक अतिरिक्त लाभ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.7: समस्या का विवरण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.8: आवश्यकता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

16.9: सारांश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 17: रेडीनेस की परिभाषा की पृष्ठभूमि

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.1: लीन का काम करने की अनुमति

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.2: कानबान के कॉलम प्रवेश मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.3: स्क्रम और अन्य एजाइल प्रक्रिया की अपोक्रीफा

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.4: सर्फिंग > कोडिंग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.5: पॉइंट सॉल्यूशंस

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

17.6: हम तैयार होने के लिए तैयार हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 18: कार्य-तत्परता परिभाषा को परिभाषित करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.1: उद्देश्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.2: अनुकूलित

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.3: संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.4: उदाहरण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.5: सहमति

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.6: प्रवेश नियंत्रण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

18.7: सारांश

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 19: तत्परता की विशिष्ट परिभाषा (RMF 3)

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.1: प्रक्रिया में एक और प्रवेश द्वार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.2: कार्य तत्परता की परिभाषा की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.3: प्रोडक्ट एग्जिट क्राइटेरिया

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.4: इंजीनियरिंग प्रवेश मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.5: दोहराव में कोई नुकसान नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.6: व्यवहार: एक या अधिक डीओआर टेम्पलेट बनाए रखें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.7: Definition of Ready टेम्पलेट क्यों रखें?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.8: गतिविधि: DoR टेम्पलेट को परिभाषित करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.9: समय के साथ DoR टेम्पलेट्स का रखरखाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.10: टेम्पलेट्स केवल एक प्रारंभिक बिंदु हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.11: व्यवहार: विशिष्ट कार्य-तत्परता की परिभाषाओं पर सहमति

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.12: गतिविधि: कार्य-मद के लिए कार्य-तत्परता की परिभाषा तय करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.13: व्यवहार: कार्यान्वयन शुरू करने से पहले आइटम को तैयार करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.14: टीम के नियंत्रण से बाहर की स्थितियां

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.15: व्यवहार: कार्य आइटम में तैयारी को ट्रैक करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.16: व्यवहार: तैयारी के आधार पर कार्य को नियंत्रित करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.17: गतिविधि: तैयारी निर्धारित करने के लिए DoR का उपयोग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

19.18: संक्षेप में

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

भाग V: संश्लेषण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 20: ज्यादातर समय सीमाएं मायने नहीं रखतीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.1: वास्तविक समय सीमाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.2: मनमानी समय सीमाएं मायने नहीं रखतीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.3: एयरलाइंस

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.4: तकनीकी कर्ज की उत्पत्ति

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.5: विपणन और बिक्री की समय-सीमाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.6: परियोजना समय-सीमाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.7: एक और रास्ता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.8: मनमानी समय-सीमाएं आवश्यक नहीं हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.9: मनमानी समय-सीमाओं को समाप्त किया जाना चाहिए

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.10: वास्तविक समय-सीमाएं अभी भी एक कारक हैं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.11: पिस एलर¹

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

20.12: निष्कर्ष

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

¹अंतिम उपाय, मूल रूप से फ्रांसीसी भाषा से।

अध्याय 21: क्षमता 1: आवश्यकता परिपक्वता प्रवाह

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.1: जैसा नीचे, वैसा ऊपर

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.2: सिद्धांत: सभी आवश्यक कार्य में पारदर्शिता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.3: व्यवहार: ज़िम्मेदारी काम के साथ चलती है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.4: व्यवहार: आवश्यकताओं की स्थिति का दृश्य ट्रैकिंग

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.5: व्यवहार: तैयारी और कार्यान्वयन से जुड़े सभी कार्यों को प्रकट करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.6: गतिविधि: तैयारी कार्य

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.7: व्यवहार: एक कार्य वस्तु को पूरा करने के लिए आवश्यक सभी कार्यों को प्रकट करें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.8: व्यवहार: समय-सीमा की बजाय तत्परता को प्राथमिकता दें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

21.9: निष्कर्ष

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 22: स्क्रम में RMF के साथ कार्य और सूचना का प्रवाह

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.1: प्रारंभिक टिप्पणियां

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.2: प्रारंभिक आवश्यकता को समझना और तैयार करना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.3: तैयारी कार्य का आरंभ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.4: तैयारी कार्य की योजना और क्रियान्वयन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.5: तैयारी परिणामों की समीक्षा

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

22.6: कार्यान्वयन की योजना और समापन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 23: RMF का प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

23.1: एक आवश्यकता का जीवन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

23.2: सूचना और कार्य का उदाहरण प्रवाह

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

23.3: पहले

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

23.4: बाद में

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

23.5: लाभ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 24: RMF की ओर संक्रमण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.1: अपनाने का पैटर्न

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.2: स्क्रम कार्यप्रवाह में स्थापना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.3: अन्य फ्रेमवर्क

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.4: प्रमुख विरोध: तैयारी कार्य आइटम

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.5: बड़ा बदलाव: मानसिकता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.6: परिवर्तन के लिए सलाह

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

24.7: निष्कर्ष

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

अध्याय 25: यह आप पर निर्भर है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

25.1: संक्षेप

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

25.2: अब, यह आपकी बारी है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

भाग VI: संसाधन

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

परिशिष्ट A: Scrum समस्या नहीं है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

Scrum क्या है?

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

फ्रेमवर्क

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

स्क्रम परियोजना- और कार्य-प्रबंधन को संबोधित करता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

Scrum को प्रोडक्ट मैनेजमेंट फ्रेमवर्क की तरह मानने की मूर्खता

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

आवश्यकताओं को परिपक्व करने के लिए कोई निर्धारित तंत्र नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

आवश्यकताओं के विकास में इंजीनियरिंग विशेषज्ञता का कोई निवेश नहीं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

एंटीपैटर्न्स

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

आवश्यक विस्तार

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

स्क्रम में जोड़ना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

परिशिष्ट B: सिनैप्स फ्रेमवर्क™

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सिनैप्स क्या कवर करता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

तीन प्रवीणताएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सिनैप्स को कैसे अपनाया जाता है

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

दो ढांचों का मिश्रण

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

साइनैप्स फ्रेमवर्क की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

संगठनात्मक प्रवीणताएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

संगठनात्मक क्षमताएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

संगठनात्मक आदतें

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सिनैप्स फ्रेमवर्क की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

एक व्यवहार-सिद्धांत की संरचना

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

क्रम का महत्व

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

इस पुस्तक पर सिनैप्स का प्रभाव

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

परिशिष्ट C: RMF 1 के प्रति सामान्य आपत्तियां और बाधाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सामान्य आपत्तियां

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सामान्य बाधाएं

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

परिशिष्ट D: डीओडी प्रारंभिक मानदंड सूचियाँ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

इंजीनियरिंग निकास मानदंड प्रारंभिक सूची

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

उत्पाद प्रवेश मानदंड प्रारंभिक सूची

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

परिशिष्ट E: डीओआर स्टार्टर क्राइटेरिया सूचियाँ

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

प्रोडक्ट एग्जिट क्राइटेरिया

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

इंजीनियरिंग प्रवेश मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

स्प्रिंट प्रवेश मानदंड

यह सामग्री नमूना पुस्तक में उपलब्ध नहीं है। पुस्तक को Leanpub पर <https://leanpub.com/ready-hi> से खरीदा जा सकता है।

सूची

- AI, 22
- API, 24
- auto-antonyms, 17
- banks, 10
- Barreras, Tom, viii
- big-batch development, 18
- C#/.NET solution, 3
- Central Oregon, vii
- confirmed shared understanding, 17
- culture, 10
- dependency services, 10
- development environment, 10
- DoD को परिपक्व करें, 51
- domain, 10
- engineering practices, 10
- engineering teams, 10
- Federated Credit Institution, 2
- finance, 10
- Fungible people, 25
- leadership, 10
- Level of Detail, 55
- loan repayment portal, 3
- low-code solution, 3
- natural languages, 17
- OutSystems, 3
- Producure, iii
- Product Backlog Item, 11
- Product Backlog Item's Definition of Done, 51
- Product organization, 4
- Product Owner, 10
- Project Management Institute, 2
- Requirements Maturation Flow, 28, 75
- requirements-authoring technique, 10
- RMF 1, 32, 87
- RMF 2, 55
- Scrum, 17, 75
- server resources, 25
- shared understanding, 36
- Sprint, 11
- Stakeholder Involvement, 55
- technical debt, 69
- test-development, viii
- United States national financial infrastructure, 2
- Waterfall, 17
- अतिरिक्त कार्य, 21
- अन्य फ्रेमवर्क, 78
- अपनाने का पैटर्न, 78
- अपव्यय, 27
- अपस्ट्रीम सर्विस, 13
- आवश्यकता परिपक्वता प्रवाह, i, iii, iv, 72
- आवश्यकताओं का परिपक्वता प्रवाह की ओर संक्रमण, 78
- आवश्यकताओं का लेखन, 16
- इंजीनियरिंग कार्यकारी, 4
- इंजीनियरिंग टीमों, 9
- इंजीनियरिंग प्रथाएं, 16
- इंजीनियरिंग प्रवेश मानदंड, 64
- एजाइल, 26

- एजाइल परिवर्तन, iv
 कस्टम तत्परता परिभाषाएं, iii
 कानबान, i
 काम से लादना, 9
 कार्य आइटम, 26
 कार्य आइटम का पूरा होना, 20
 कार्य कब पूरा हुआ यह कैसे जानें, 24
 कार्य तत्परता की परिभाषा की संरचना, 64
 कार्य समझौता, 55
 कार्य समाप्ति की परिभाषा, 50
 कार्यप्रवाह, 36, 50
 कुछ तो कम है, 1
 कोड गुणवत्ता, 27
 कोडबेस, 27
 गलत आवश्यकताएं, 18
 गार्बेज-इन, गार्बेज-आउट, 16
 गेटिंग कार्यान्वयन, 57
 गोल्ड प्लेटिंग, 22
 जवाबदेही और अनुरेखण, 15
 जीत को हार में बदल देना, 19
 जोखिम, 20
 ज्ञान-आधार, 8
 ज्यादातर समय सीमाएं मायने नहीं रखतीं, 69
 टर्मिनेटर-शैली के विद्रोह, 22
 टीम कौशल, i
 टुकड़ों की गिनती, 23
 डिज़ाइन पैटर्न, iv
 डेटा की हानि या भ्रष्टता, 21
 डेफिनिशन ऑफ़ डन, 24
 तत्परता की विशिष्ट परिभाषाएं, 64
 तैयार, 13
 तैयारी कार्य आइटम, 35
 तैयारी के लिए स्थान बनाना, 31
 द बिग टर्नअराउंड, 13
 द बैंक, 13
 दोष, 26
 धारणा, 26
 नियामक उल्लंघन, 22
 नेतृत्व टीम, 7
 परियोजना समय-सीमाएं, 70
 परीक्षक, 13
 पहेली बनाना, 23
 पीकेबी-संचालित विकास, iv
 पीछे समय-सारिणी से, 27
 पुनरावृत्ति, 24
 पुनर्काम, 18
 पुरानी प्रणाली, 7
 पूर्ण की परिभाषा, 21
 पूर्णता का अनुसरण, 51
 प्रक्रिया में प्रवेश द्वार, 64
 प्रतिक्रिया चक्र, 14
 प्रतिभा प्रतिधारण, 27
 प्रमुख विरोध, 78
 प्रवाह, 14
 प्रारंभिक कार्य, 55
 प्रोडक्ट एग्जिट क्राइटेरिया, 64
 प्रोडक्ट ओनर, 5, 21
 प्रोडक्ट और इंजीनियरिंग, 15
 प्रोडक्ट बैकलॉग आइटम, 8, 13, 15, 23
 प्रोडक्ट मैनेजर, 5
 प्रोडक्ट, भूमिका, 4
 बर्बाद समय, 22
 बाज़ार हिस्सेदारी, 22
 बैकलॉग आइटम, 27
 मध्य-बीसवीं सदी के प्रोग्रामिंग सिद्धांत, 7
 मध्य-स्प्रिंट परिवर्तन, 6
 मौजूदा कोड, 12
 रेडी, i, ii
 रोलआउट, 55
 वास्तविक आवश्यकता, 21
 विकास टीम, 20
 विपणन और बिक्री, 70
 विशिष्ट कार्य-समाप्ति परिभाषाएं, 49
 विश्वास, 27
 विषय विशेषज्ञ, 7, 12
 व्यक्तिगत योगदानकर्ता, 5
 व्यवहार-संचालित विकास, iv, 8
 व्यवहार: आवश्यकताओं की स्थिति का दृश्य ट्रेकिंग, 73
 व्यवहार: तैयारी और कार्यान्वयन से जुड़े सभी कार्यों को प्रकट करें, 73
 संचयी लागतें, 26
 संसाधन, 81

सफलता का दावा करना, 39
समय-सीमा, 24
सलाहकार, 8
सिस्टम, 10
सिस्टम डाउनटाइम, 21
सुरक्षा कमज़ोरियां, 21
सॉफ्टवेयर डेवलपर्स, 21

सॉफ्टवेयर विकास, 16, 20, 23
सॉफ्टवेयर विकास तकनीकें, 7
स्क़म, 5, 24
स्क़म कार्यप्रवाह में स्थापना, 78
स्पष्ट तैयारी कार्य, 35
स्प्रिंट, 16, 20, 24, 26
हितधारक, 18