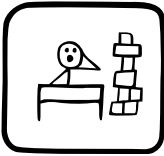


Die acht Entwicklungs-Verschwendungen

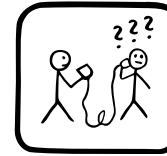
Version 0.4, Copyright © 2011 Mark Seuffert, Schweden



Featuritis

1. Funktionalität die selten benutzt, ohne Bedarf oder unter falschen Annahmen hinzugefügt wird. Das passiert wenn Dinge gemacht werden die eigentlich keiner haben will, statt Zusammenarbeit mit Kunden und Endbenutzern.

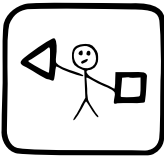
Was man tun kann: Es ist wichtig Ziele, Kontext und Prioritäten bei der Entwicklungsarbeit zu kennen.



Informationsverlust

5. Wissen das verloren geht durch ungeeignete Kommunikation oder wechselnde Teams. Dies passiert wenn Gespräche durch Dokumente ersetzt werden oder Entscheidungen nicht aufgeschrieben werden.

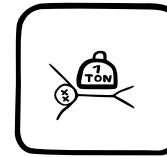
Was man tun kann: Schaffe einfachen Zugang zu Experten und Informationen.



Unstimmigkeit

2. Extraarbeit durch ähnliche Dinge die unterschiedlich erledigt werden. Es kostet Zeit zu verstehen wie Teile zusammenpassen, umso mehr wenn diese noch verändert oder angepasst werden müssen.

Was man tun kann: Im Zweifelsfall setze auf Konsistenz, benutze immer die gleichen Muster und Standards.



Technische Schulden

6. Extraarbeit angesammelt durch Entwicklungsmängel. Ein Fokus auf kurzzeitige Ergebnisse erhöht Komplexität, vermindert Qualität, was wiederum zu Mehraufwand und komplettem Neuschreiben führt.

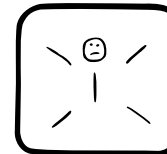
Was man tun kann: Vernachlässigtes Design ist teures Design, investiere in Pflfegbarkeit und Refactoring.



Wartezeit

3. Behinderung durch Wartezeiten und Unterbrechungen. Das Warten auf Aktivitäten und Menschen verhindert weitere Aufgaben fertig zu stellen, zusätzlich wirkt sich ein ständiges Stop-und-Start negativ auf Produktivität aus.

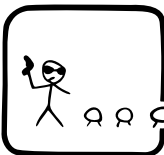
Was man tun kann: Visualisiere Arbeitsfluss, zeige wo Engpässe und teilweise erledigte Arbeit auftreten.



Zerstreung

7. Behinderung durch Aufteilung mit langen Kommunikationswegen. Die Gesamtproduktivität sinkt mit spezialisierten oder verteilten Teams, es ist besser wenn Wissensgebiete eng zusammenarbeiten.

Was man tun kann: Vermeide Sub-Optimierung und bringe engagierte Menschen zusammen.



Fehlersuche

4. Aufwand zum Erkennen und Beseitigen von Fehlern. Es ist teurer Fehler erst spät in der Entwicklung zu finden, da die Beseitigung schwerer ist wenn Details nicht mehr bekannt oder mehr Personen betroffen sind.

Was man tun kann: Statt Massentests, baue Qualität von Anfang an ein und halte regelmäßige Reviews.



Einengende Struktur

8. Umgebung die schwer zu verändern ist oder zu wenig Unterstützung bietet. Warnsignale sind schwergewichtiges Management, einschnürende Prozesse, ungenutztes Talent und unglückliche Angestellte.

Was man tun kann: Ein gesundes System kann sich angleichen, lernen und fortlaufend verbessern.

Definition: Verschwendung ist alles was man aus einem System entfernen kann um die gleichen (oder bessere) Ergebnisse zu erhalten. Tipp: Finde heraus was auf lange Sicht Wert erzeugt und was nicht, entferne Überflüssiges und Unproduktives.

Geschichte: Die Kategorisierung von Verschwendung (Japanisch "Muda") hat ihren Ursprung in Lean Manufacturing (Shingo & Ohno) und Lean Software (Poppendiek). Das Agile Manifesto kennt das Prinzip der Maximierung von nicht getaner Arbeit.