

Eliminate Waste

Wie wir unseren Weg aus dem Backlog-Chaos gefunden haben.



Vorstellungsrunde

Produkt A:

- 200k LOC (klein),
- interne Qualität gut,
- hohe fachliche Komplexität,
- 6 Jahre in Development
- Am Ende seines Development Lifecycles

Produkt B:

- 35k LOC (kleiner)
- interne Qualität mittelmäßig
- hohe fachliche Komplexität
- 2 Jahre in Development
- Am Ende seines Development Lifecycles



Ich (Moritz):

- 36 Jahre alt, davon 10 Jahre bei andrena.
- Ehemaliger Mathematiker & Hobby-Programmierer.
- Agiler Software-Techniker.
- Im neuen Team als Technical Coach & Ausbilder.

Symptom

Wir haben 2 Backlogs:

- Main Backlog: Ca. 70 User Stories. Wenig fachliche Kohäsion.
- Ideenpool: Ca. 60 User Stories. Wurde mal angelegt als das Main Backlog zu groß wurde, um Stories woanders zu ‚lagern‘.

Waste:

- Partially done work
- Relearning
- Waiting
- Software Bloat



Ursache – No Direction

Kein Product Goal

- Kein klares Ziel, was gerade erreicht werden soll.
- Keine Messungen.
- Kein erkennbarer Fortschritt.

Gab einen fehlgeschlagenen Versuch ein Product Goal zu etablieren:

Zusätzliches mobile App Frontend mit neuen extra für mobile-Funktionalitäten.

- => Viel Designworkshops & fachliche Ausarbeitung.
- => Wurde von Führungskräften (spät) unterbunden.
- => Grund: Das Produkt soll nicht komplexer werden.



Verchromtes Fahrrad

Lösung

Just try again:

- Erster Versuch schlug fehl
- Wir wissen wieso
- Nächster Versuch wird besser

Was wir wissen:

- Das Produkt soll nicht komplexer werden.
- Das Produkt ist am Ende seines Lifecycles.
- Das Produkt soll ‚fertig‘ werden.

~~„Was können wir als nächstes Tolles bauen?“~~

- ~~• Motivierend, wenn etwas ‚Tolles‘ gefunden wird.~~
- ~~• Hohes Risiko von Software Bloat.~~

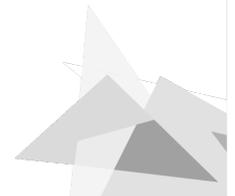


„Was fehlt mit wir ‚fertig‘ sind?“

- Less fun. Kein *fancy new stuff*.
- Niedriges Risiko Dinge zu bauen ohne Nutzen.
- Motivierend, wenn es Sinn macht.
- Findet Unterstützung auf den Leitungsebenen.

Neues Product Goal:

Betrieb & Wartung können mit minimalem Personenaufwand geleistet werden



Symptom

Was ist besser geworden?:

=> Wir wissen, wofür wir da sind, was wir eigentlich machen wollen.

Was ist immer noch schlecht?:

- 50% der Inhalte der Sprints Zahlen nicht auf ein Product Goal ein
- Alles wird nur langsam fertig
- Ein feature branch lebt schon seit Monaten
- Irgendwie zahlt jede Story auf etwas anderes ein.

Waste:

- Task Switching



Ursache - Legacy Entscheidungen

Generell:

- Als noch No-Direction herrschte, wurden viele Entscheidungen getroffen, die noch leben.
- Große Dinge, die man sich nicht getraut hat zu machen und auch nicht getraut hat nicht zu machen, werden nach und nach umgesetzt. Eine Story alle 1-2 Sprints.

Konkret:

- Entwickler haben große Refactorings ausgehandelt, sind unglücklich wenn sie die nicht fertig machen dürfen.
- Story – „Änderung am Berechtigungssystem“ zieht sich. Geschätzte 8 SP sind in Wirklichkeit eher 40 SP.
- Migration in die cloud wird nach und nach auf einem langlebigen feature-branch weitergebaut.

Lösung

- Entwickler haben große Refactorings ausgehandelt, sind unglücklich wenn sie die nicht fertig machen dürfen.
=> Renegotiate.
- Story – „Änderung am Berechtigungssystem“ zieht sich. Geschätzte 8 SP sind in Wirklichkeit eher 40 SP.
=> Fokus.
- Migration in eine neue Plattform wird nach und nach auf einem langlebigen feature-branch weitergebaut.
=> Minimum Viable Migration von NonProd definieren.

Neue Regel: **Stop starting**. Get things **DONE**.

Neue Regel: **Do or do not**.

Symptom

Was ist besser geworden?:

- Nebenläufer-Stories sind stark reduziert.
- Sprintziele zahlen auf Product Goal ein.
- Fachliche Kohäsion der Stories im Sprint ist deutlich besser.

Was ist immer noch schlecht:

- Product Backlog ist immer noch riesig und unübersichtlich.
- Viele Dinge landen auf ‚dem Backlog-Friedhof‘ obwohl sie einen klaren Mehrwert haben und sehr leicht umsetzbar sind.

Waste:

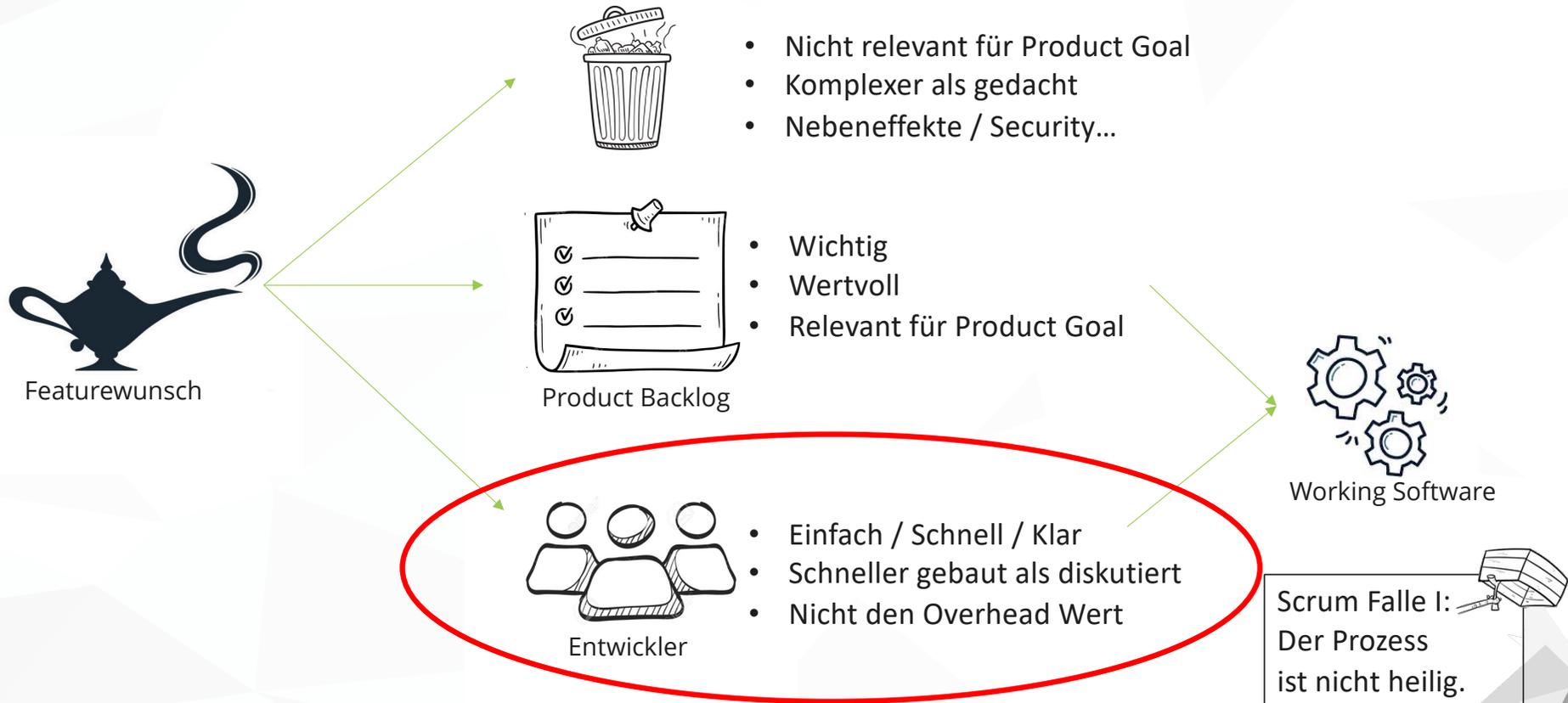
- Partially Done Work.
- Relearning.
- Waiting.



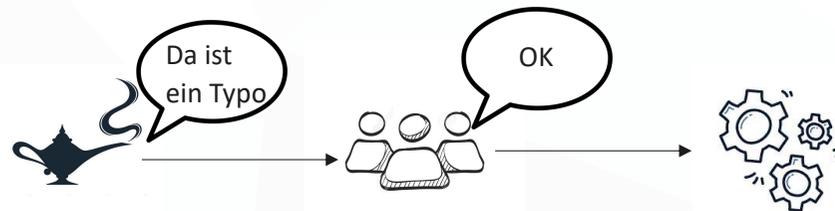
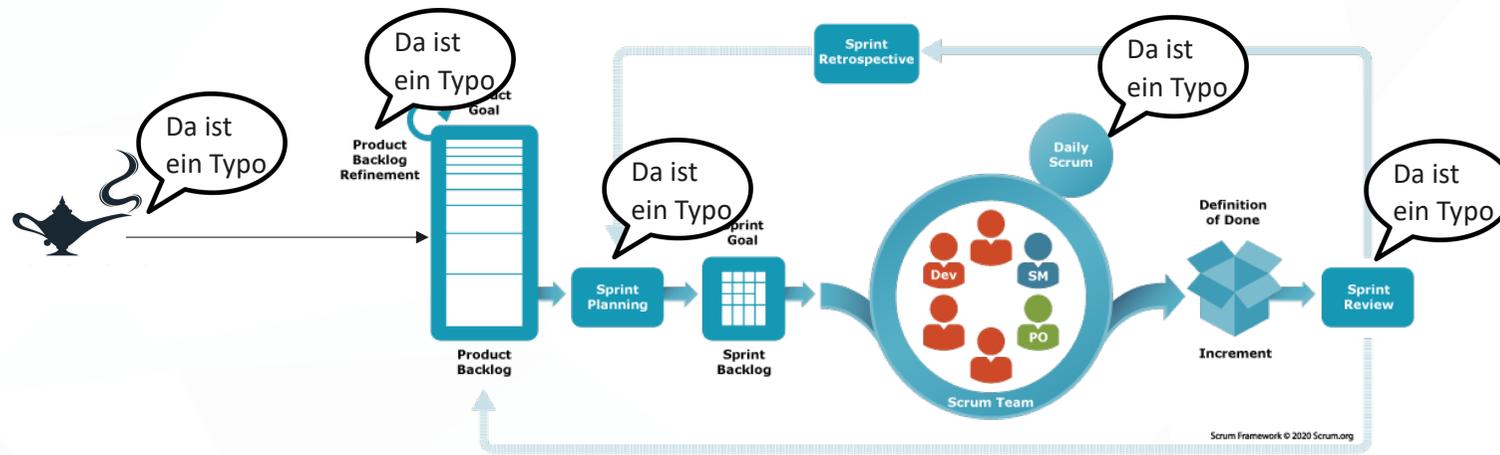
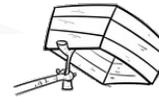
Ursache – Stakeholder (Mis-)management.



Lösung – Stakeholder Management.



Scrum Falle I: Der Prozess ist nicht heilig.



[...] we have come to value:
Individuals and interactions over
processes and tools

Symptom

Was ist besser geworden?:

⇒ Unser Backlog-Friedhof schrumpft. Stakeholder warten nicht mehr eine Ewigkeit.

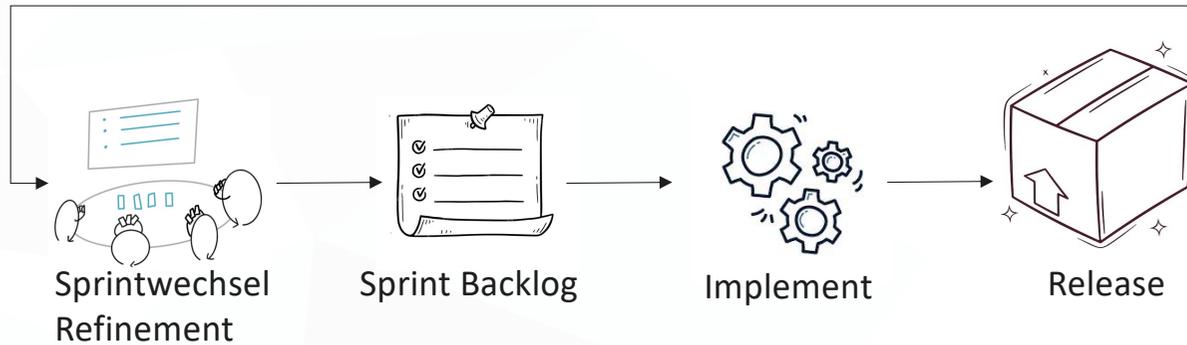
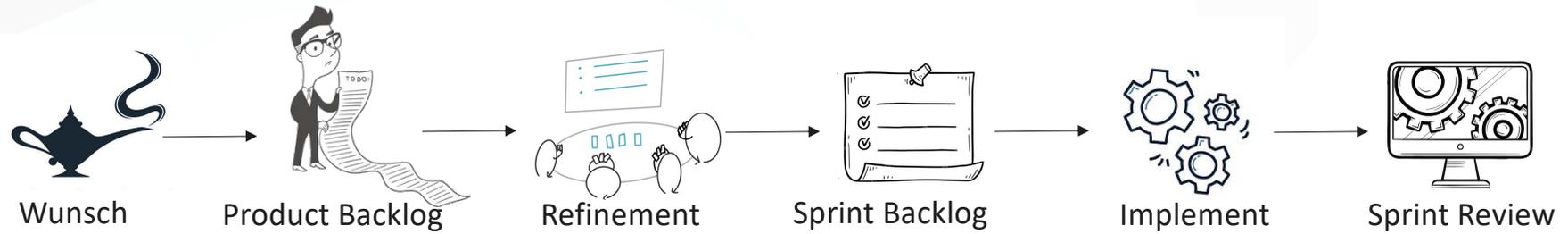
Was ist immer noch schlecht?:

- Irgendwie machen wir immer noch viele Dinge gleichzeitig.
- In Sprint Reviews kommen oft Nacharbeit-Stories auf.
- In unserer Backlog-Visualisierung wandern Themen öfters von

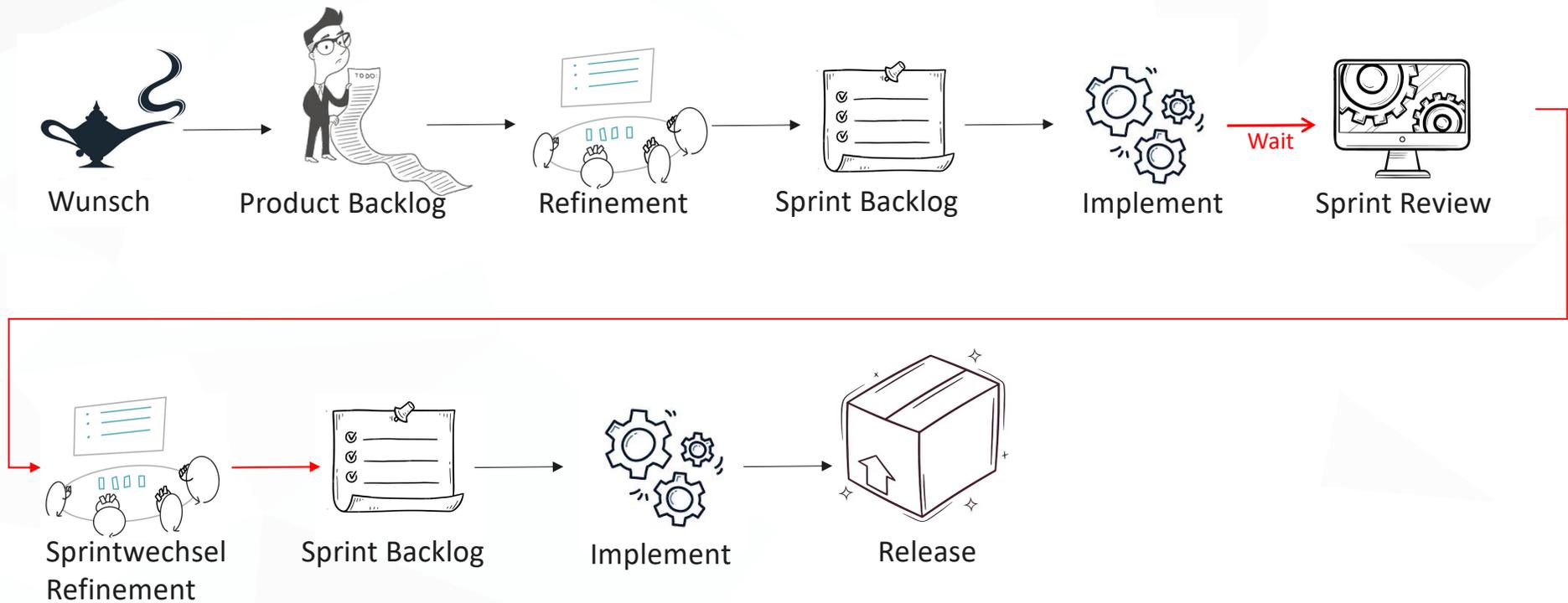
DONE -> **DOING** -> **DONE**.



Ursache – Versteckte Wartezeit



Ursache – Versteckte Wartezeit



Lösung – Definition of Done

Defintion of Done:

- [Codequalität]
- Manueller Test
- [Codequalität]
- [Codequalität]
- [Codequalität]



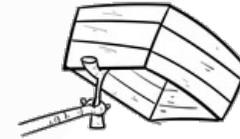
Defintion of Done:

- Stakeholder Feedback wurde eingeholt
- [Codequalität]
- Manueller Test
- [Codequalität]
- [Codequalität]
- [Codequalität]

Scrum Falle II:
DoD ist nicht nur
für Codequalität.



Scrum Falle II: Definition-of-Done ist nicht nur für Codequalität



The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

Internal Quality.

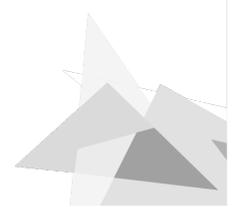
- Test Coverage
- OO
- Code Review
- ...

External Quality.

- UI & UX
- Performance
- Bugs
- ...

Zusammenfassung

| Symptom | Ursache | Lösung |
|---------------------------|-------------------------|--------------------------|
| Monster-Backlog | No Direction. | Product Goal. |
| Everything, Everywhere | Legacy Entscheidungen. | Renegotiate. Fokus. MVP. |
| Monster-Backlog II | Stakeholder Management. | Stakeholder Management. |
| Everything, Everywhere II | Versteckte Wartezeit. | Definition of Done. |



Zusammenfassung

| Symptom | Ursache | Lösung |
|---------------------------|-------------------------|--------------------------|
| Monster-Backlog | No Direction. | Product Goal. |
| Everything, Everywhere | Legacy Entscheidungen. | Renegotiate. Fokus. MVP. |
| Monster-Backlog II | Stakeholder Management. | Stakeholder Management. |
| Everything, Everywhere II | Versteckte Wartezeit. | Definition of Done. |

War das alles?

| Symptom | Ursache | Lösung |
|----------------------------|----------------------------|-----------------------------|
| Drowning in Information | Viel Input. | Visualisierung. Abstraktion |
| Security OMFG | Keine Verantwortung. | Eskalation. DevSecOps. |
| Leeres Sprint Review | Irrelevanz. Intransparenz. | Agile Coaching. |
| Everything, Everywhere III | Bottleneck PO. | Delegation. |
| Seid Ihr nicht ‚fertig‘? | Intransparenz. | Quartalsreview. |
| Architektur??? | Keine Zeit. Keine Ahnung. | Free Wednesday. |
| Revert-Undo Waste | Spontaner Chaos-Code. | Technical Coaching. |

Fazit – Scrum vs. Lean – Was ist besser?

Das finde ich toll



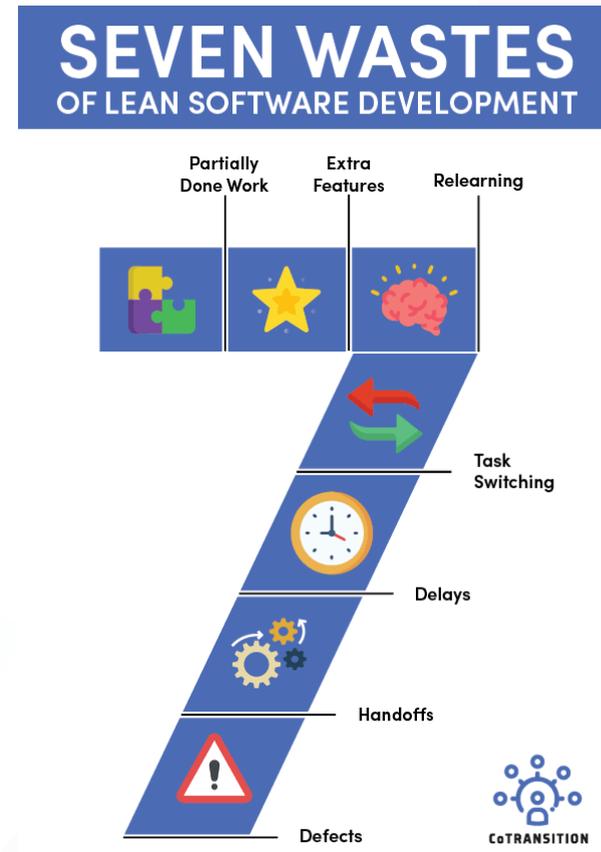
Das auch



Fazit – Die Methodik

Symptom -> Problem/Waste -> Lösung

In order to eliminate waste, one should be able to recognize it.



Lesematerial

