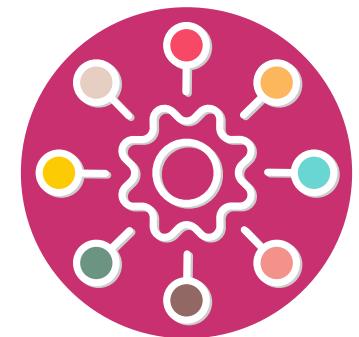
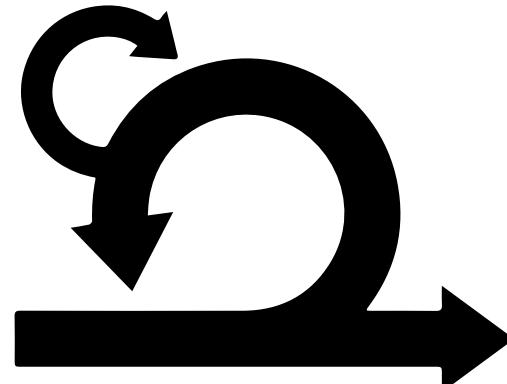




System Design Technical Management

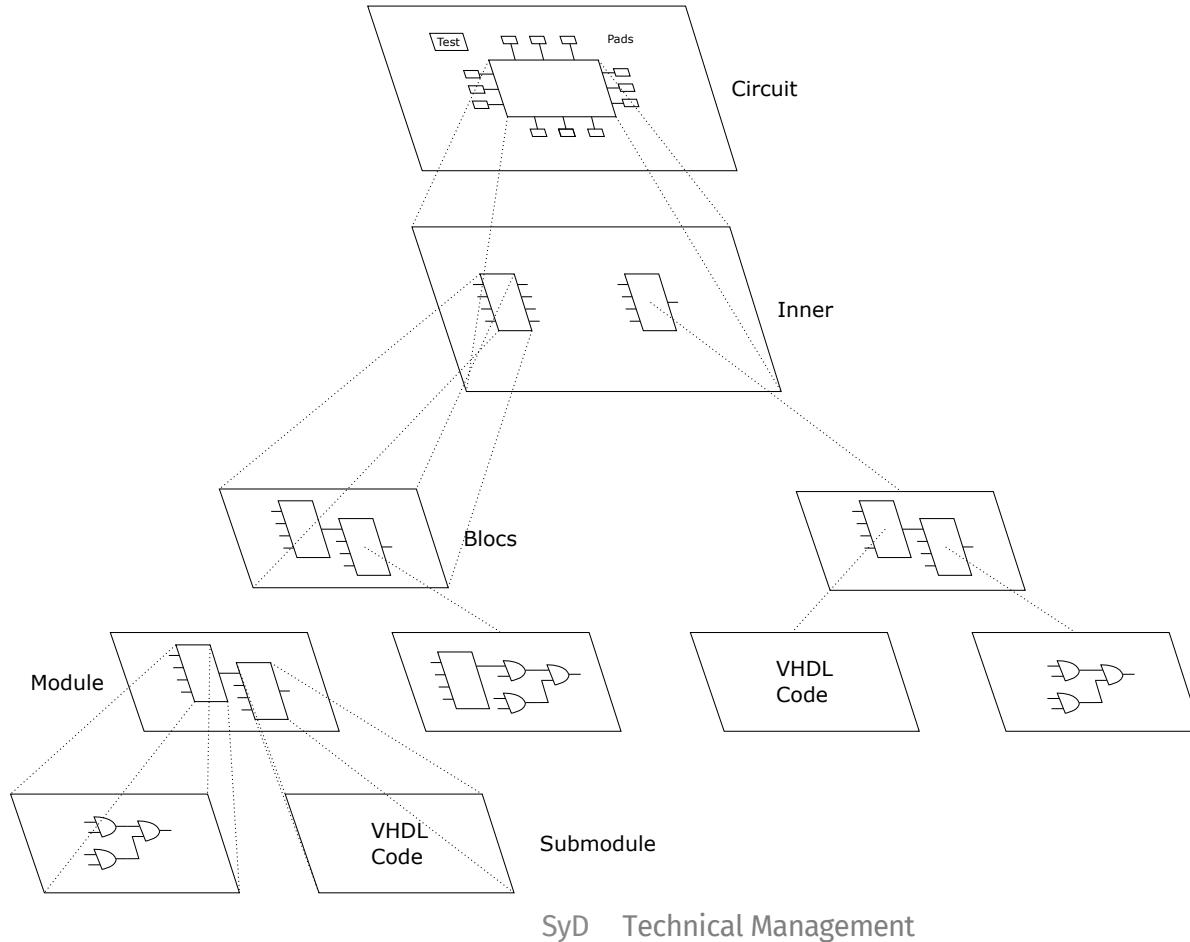
Systems Engineering program

Silvan Zahno silvan.zahno@hevs.ch





What for and why?





Technical Management Methods

- PRINCE2
- SixSigma
- Lean
- Critical chain project management (CCPM)
- Waterfall / Diagram-V / 6q
- Extreme programming (XP)
- Scrum / Kanban



Technical Management Methods

- PRINCE2
 - SixSigma
 - Lean
 - Critical chain project management (CCPM)
- Industry
-
- Waterfall / Diagram-V / 6q
- Development
-
- Extreme programming (XP)
 - Scrum / Kanban
- Software

Technical Management Methods

PRINCE2

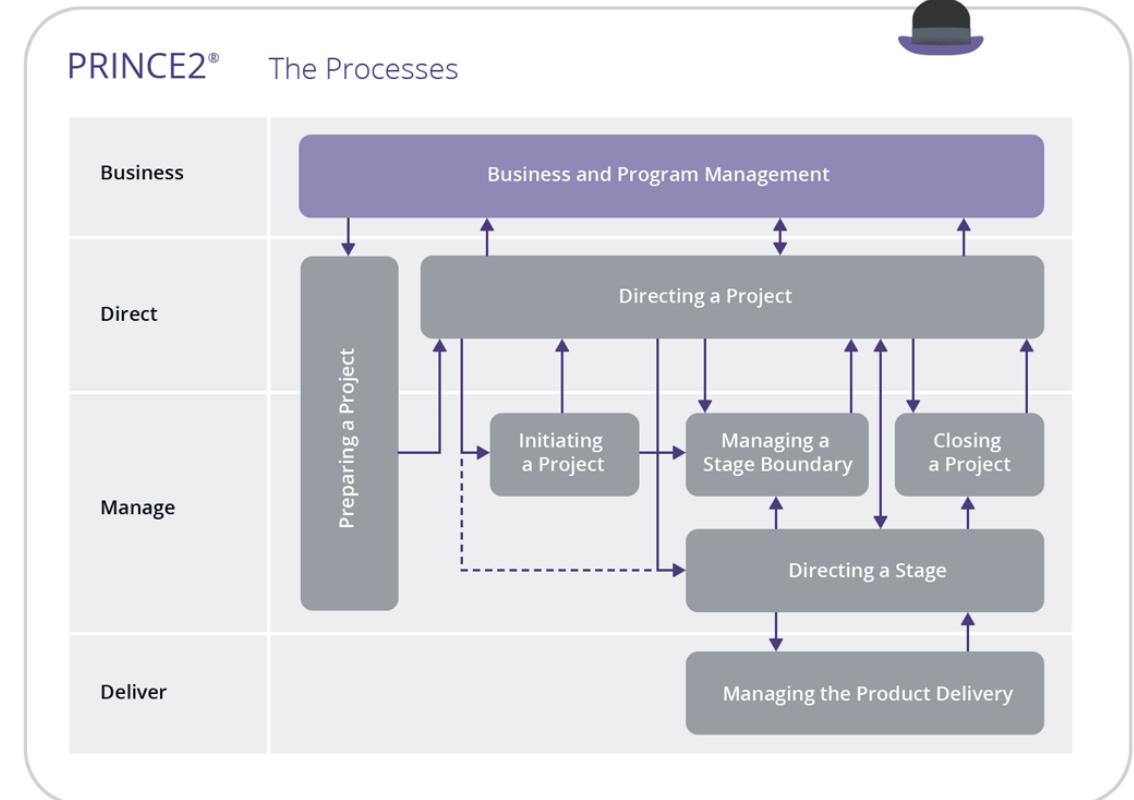
Organized approach to achieving project goals within constraints.

What it is:

Used for quality management to eliminate project defects.

Who should use it:

Large teams that tackle complex projects



<https://www.microtool.de/en/knowledge-base/how-does-prince2-work/>

Technical Management Methods

Six Sigma



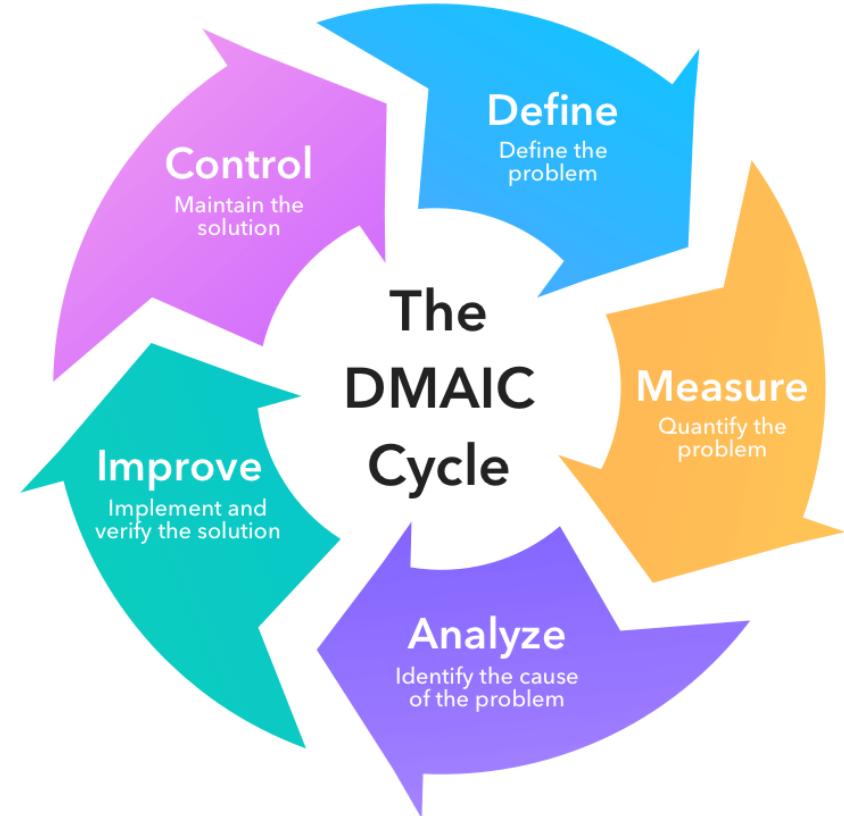
Six Sigma is a method for improving process quality by reducing variability and eliminating defects.

What it is:

Used for quality management to eliminate project defects.

Who should use it:

Large organizations looking to reduce errors



<https://startinfinity.com/project-management-methodologies/six-sigma>



Technical Management Methods

Lean (Production)

Lean is a methodology for optimizing process efficiency and reducing waste by focusing on adding value from the customer's perspective and continuously improving the process flow.

What it is:

Aims to cut waste and create a simple framework for project needs

Who should use it:

Teams struggling with efficiency issues



<https://www.sketchbubble.com/en/presentation-lean-project-management.html>

Technical Management Methods

Extreme Programming (XP)



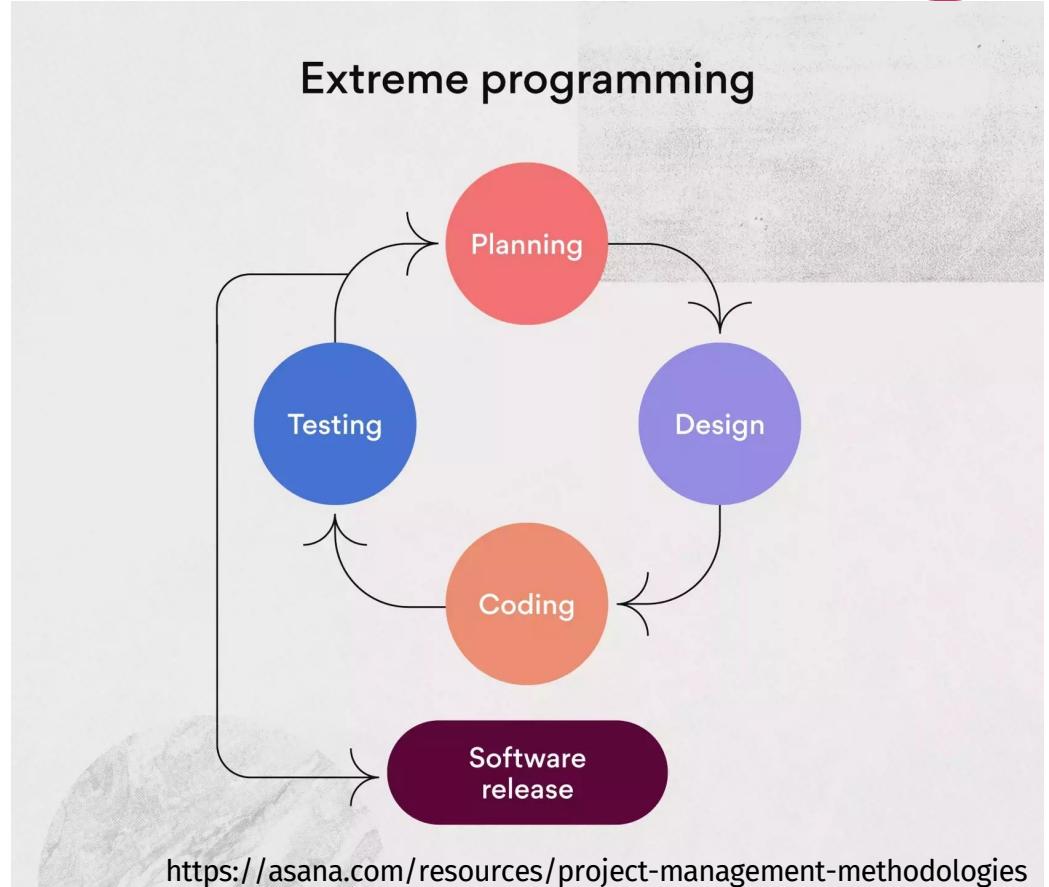
Extreme Programming (XP) is an agile software development approach that focuses on frequent releases, continuous testing, and customer feedback to deliver high-quality software products.

What it is:

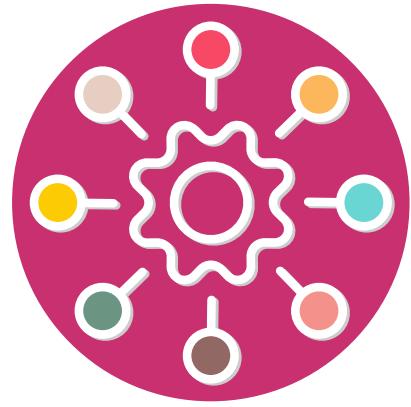
Works by creating short development cycles with many releases

Who should use it:

Small top mid-size teams that have tight turnaround times



<https://asana.com/resources/project-management-methodologies>



Waterfall / Diagram-V / 6q

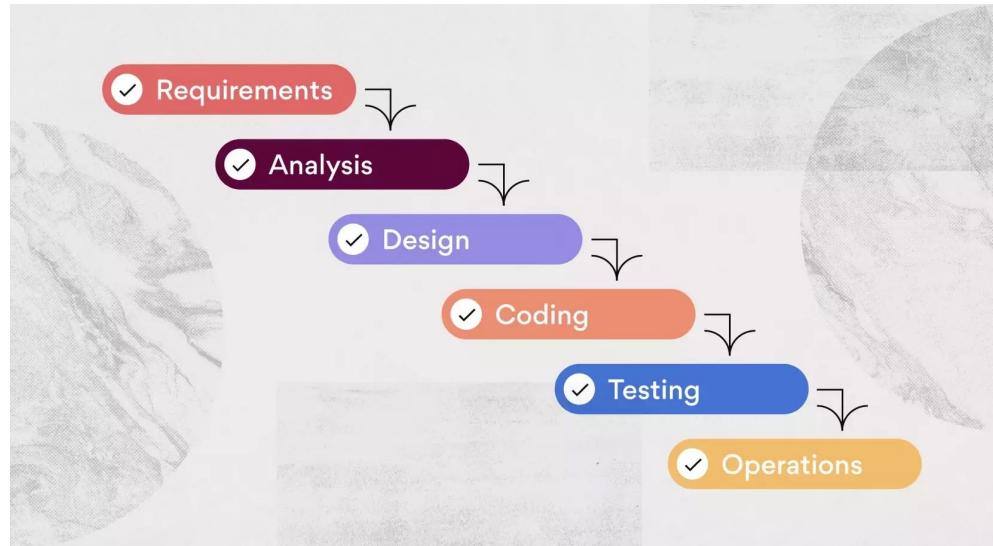
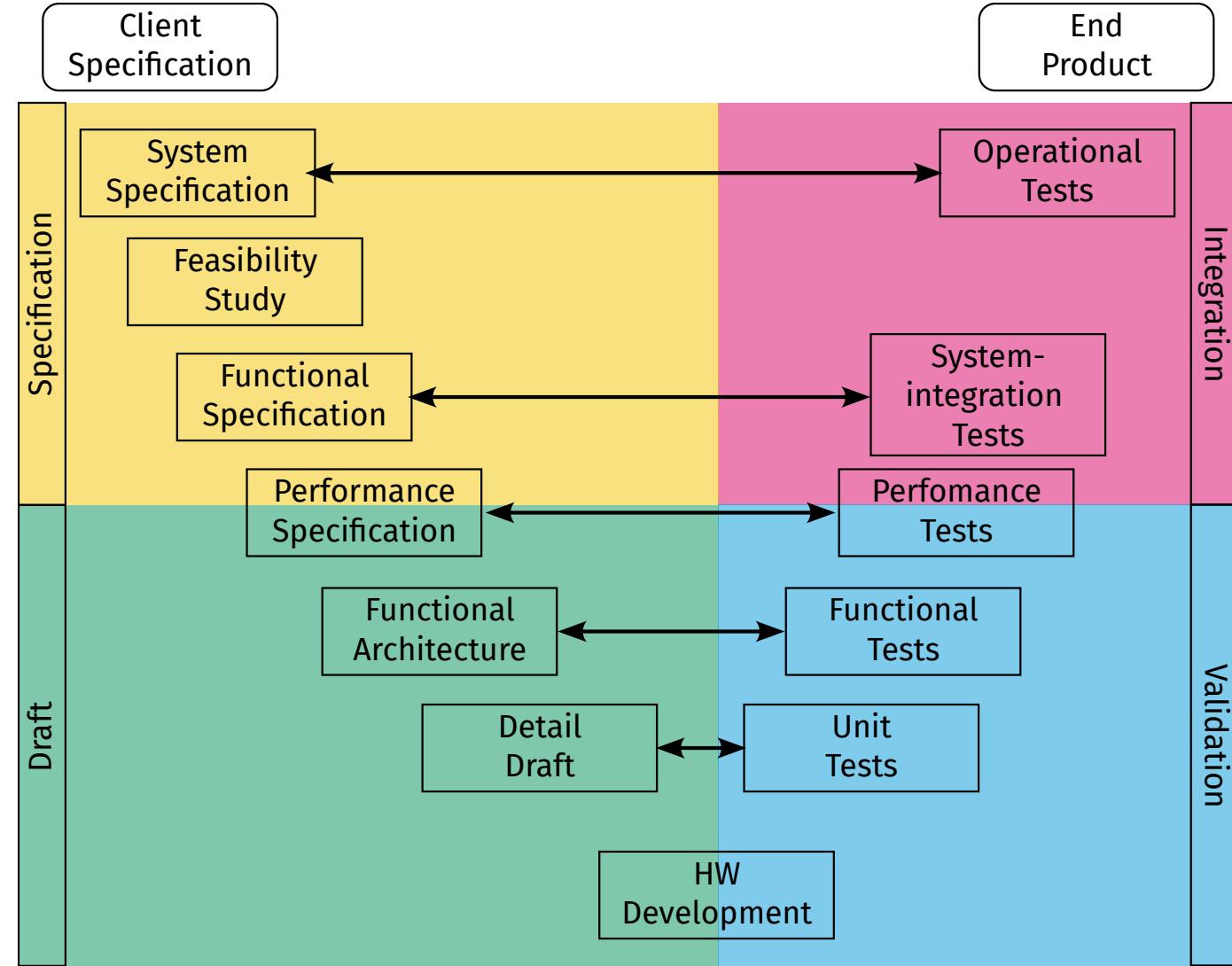
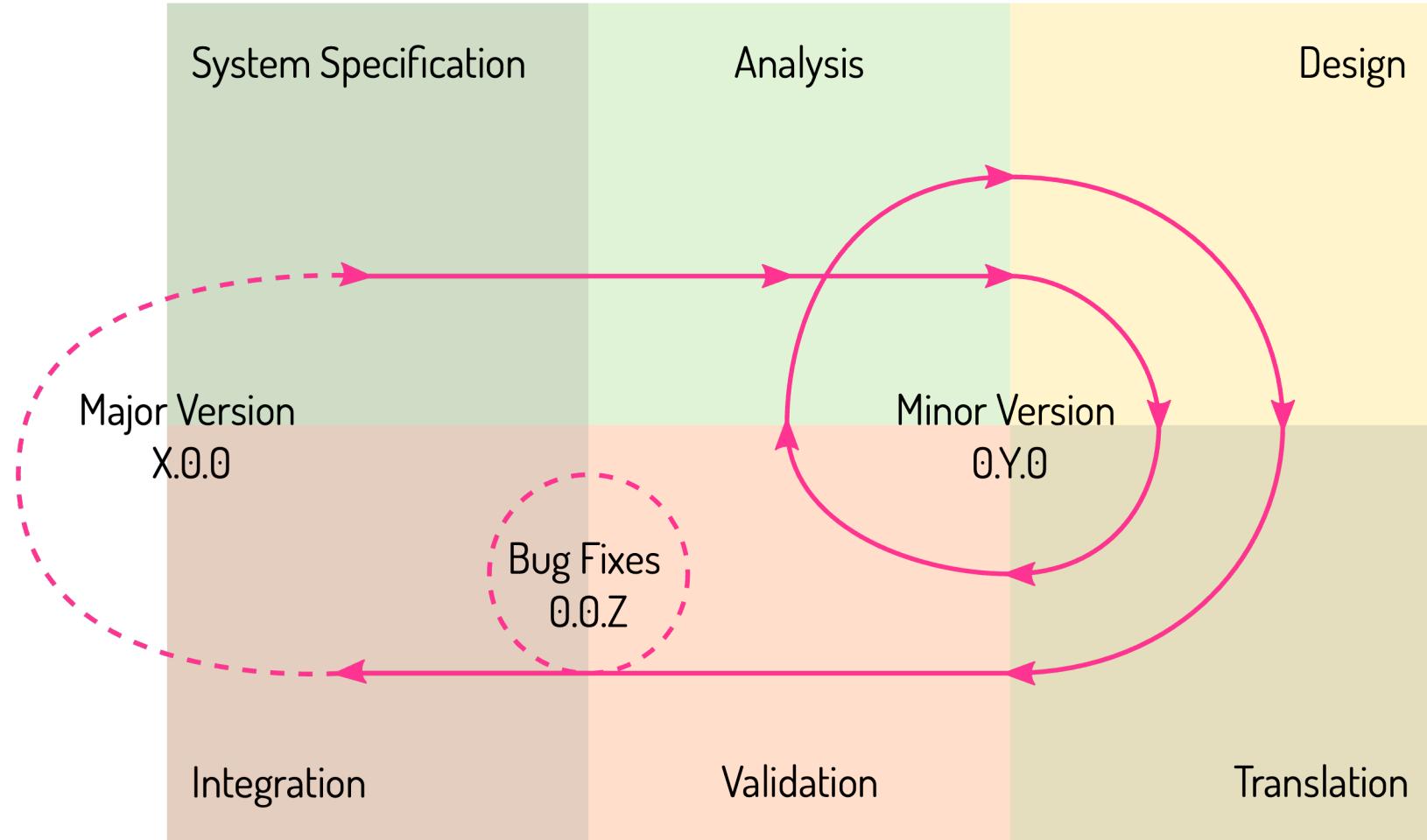
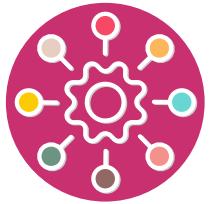
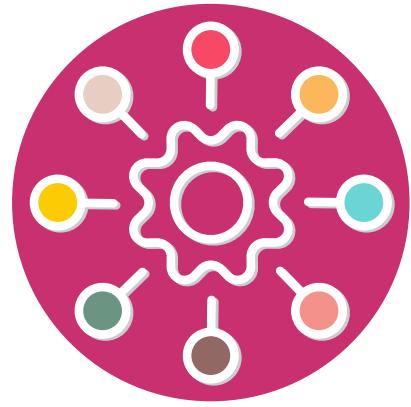


Diagram-V



6q



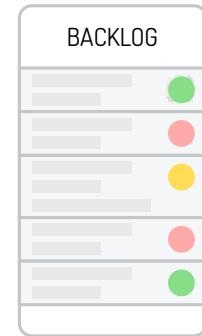


Agile
Scrum / Kanban

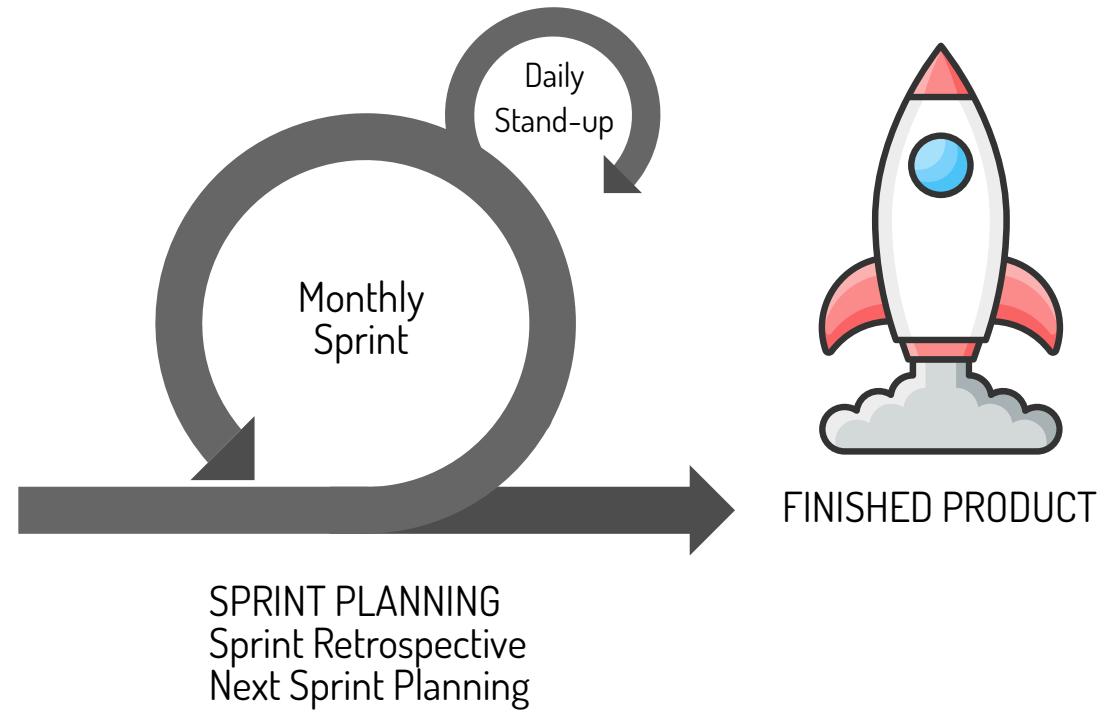
Scrum



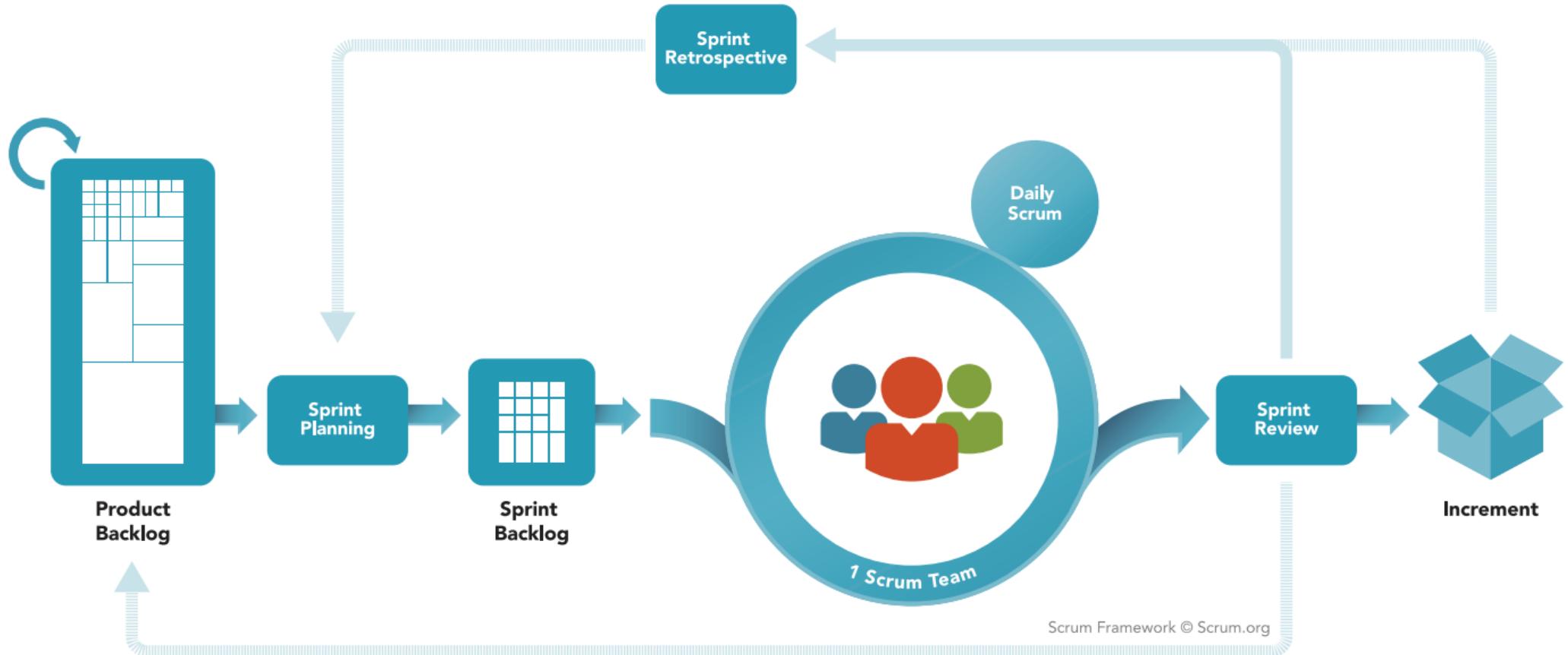
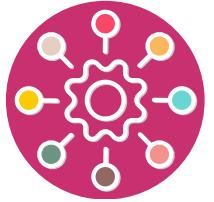
IDEA / VISION
Business Model
Milestones
Releases



PRODUCT BACKLOG
Project specific
prioritized workpackages



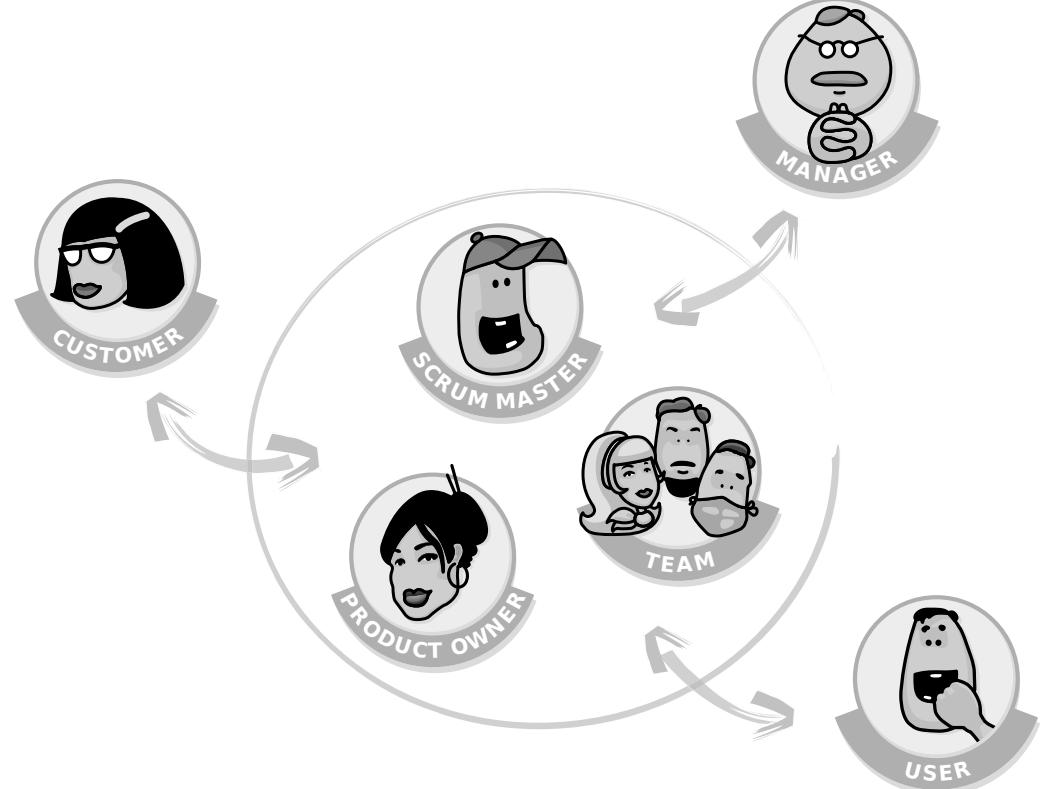
Scrum



Practical Scrum

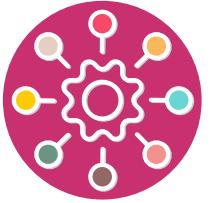


- The Roles in the Scrum Team
 - Product Owner
 - Development Team
 - Scrum Master
- The Scrum artefacts
 - User Story
 - Product Backlog
 - Sprint Backlog
 - Increment
 - Definition of Done, Definition of Ready



B. Gloer and K. Schwaber, *Scrum: Produkte zuverlässig und schnell entwickeln*, 4.,
Überarb. Aufl. München: Hanser, 2013.

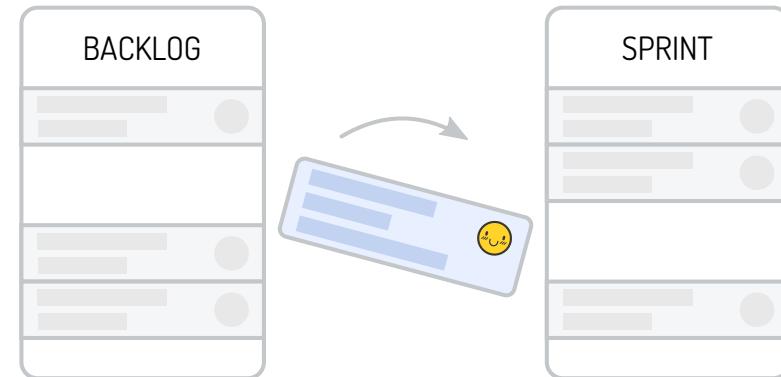
Scrum Events



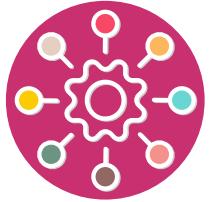
Sprint Retrospective



Sprint Planning



User Story



Explicit Title

- e.g. for a Taxi ride app: "Search for an available Taxi"

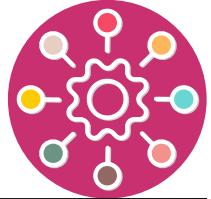
Narrative phrase: "As [persona], I want [feature], so that [function"]

- e.g. "As a customer, I want to see all nearby taxis, so that I can select one"

Acceptance Criteria

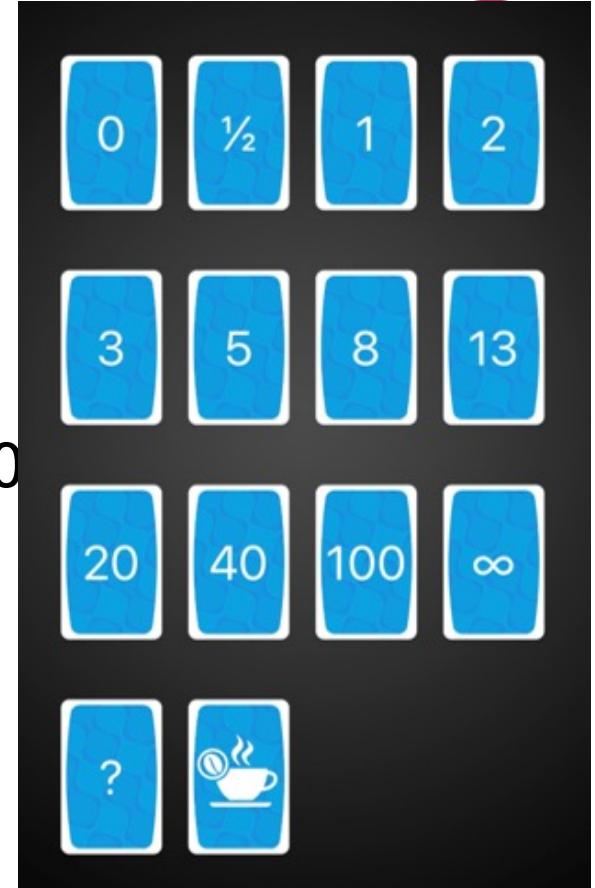
- e.g. "Only display available taxis"
- e.g. "Display them on a list from the closest to the furthest away"
- e.g. "Show a picture of the car"

Task Estimation



Scrum Time App

- Within the team 1 points needs to be defined and refined.
- Sequences
 - Planning Poker: 0, $\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 20, ..., 100
 - Fibonacci: 0, 1, 2, 3, 5, 8, 13, 21, 144
 - Natural: 1, 2, 3, 4, 5, 6, ..., 16
 - T-Shirt: XS, S, M, L, XL



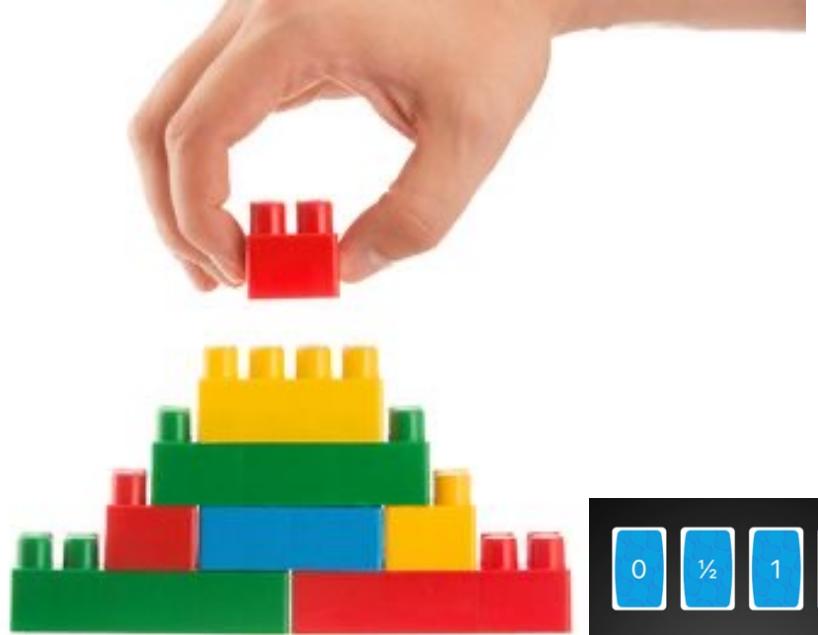


Task Estimation

Example 1

Search a Lego in a Pile and
stack is corresponds to:

1/2 Point



0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	∞
?	☕		



Task Estimation

Example 2

Create a small Lego robot

5 Points

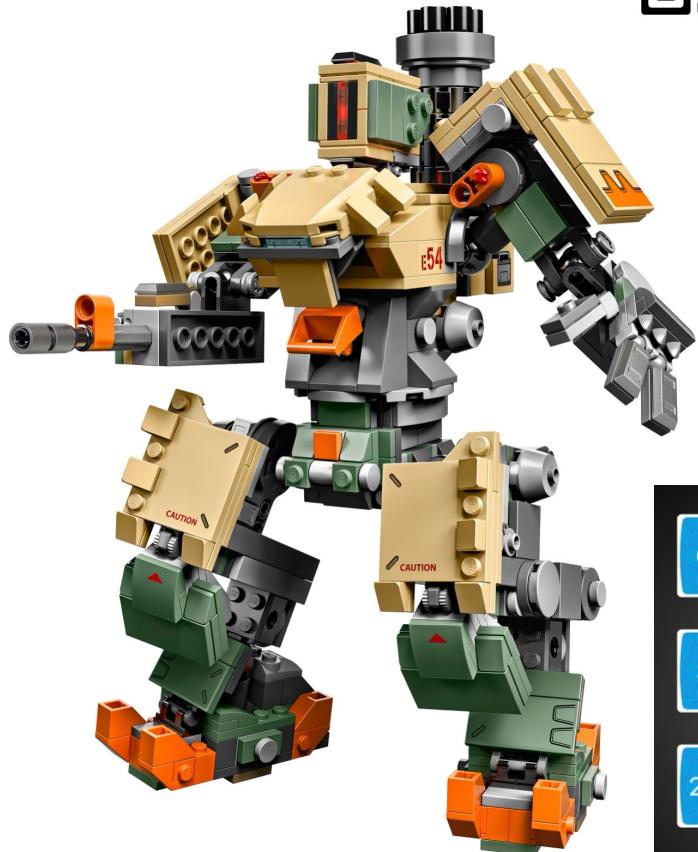


0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	∞
?	☕		

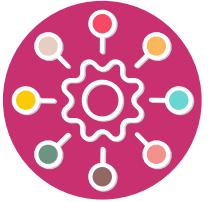
Task Estimation

Estimation 1

Create a big Lego robot
X Point(s)



0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	∞
?	☕		



Kanban

OctoArcade Invaders

Planning Sprint Board Alpha Roadmap My work Features Priority By person Status Board By status By Sprint Done

Filter by keyword or by field

Not Started (19) Estimate: 37

- planning-tracking-demo #810 Beta go-no-go meeting
- planning-tracking-demo #800 Save score across levels
- planning-tracking-demo #784 Interviews with media outlets **epic**
- Draft Enable for teams
- planning-tracking-demo #1161 tweak difficulty
- planning-tracking-demo #1167 Update README.md
- Draft Prevent the Konami code from bringing down all of GitHub

Planning (19) Estimate: 109

- planning-tracking-demo #823 Updates and bug fixes to engine from Beta **bug demo**
- planning-tracking-demo #824 Beta signup page **need help**
- planning-tracking-demo #806 [Tracking] Upsell / Growth experience **backlog feature**
- planning-tracking-demo #818 Account subscription design
- planning-tracking-demo #828 Acquire domain for launch
- planning-tracking-demo #832 Final creative shots from game
- planning-tracking-demo #829 Updates to velocity of the ship and alien movements

Building (8) Estimate: 40

- planning-tracking-demo #1160 Update documentation
- planning-tracking-demo #814 Updates to collision logic **enhancement**
- planning-tracking-demo #816 Free and paid levels **need help**
- planning-tracking-demo #831 Documentation and Support **need help**
- planning-tracking-demo #821 Updates to alien, beam, bomb and cannon sprites **#370**
- planning-tracking-demo #802 Updates to velocity of the ship and alien movements

Review (5) Estimate: 17

- planning-tracking-demo #822 Hero site - Development **#12 #1160 in-review task urgent**
- planning-tracking-demo #808 General bug fixes from Alpha feedback **#992**
- planning-tracking-demo #1151 Design new launch screen **#374 web**
- planning-tracking-demo #793 Polished alien, beam, and cannon sprite files
- planning-tracking-demo #1101 [Tracking] Integrate payments system **backlog feature**

+ Add item

<https://github.com/features/issues>

DoR – Definition of Ready (Example)

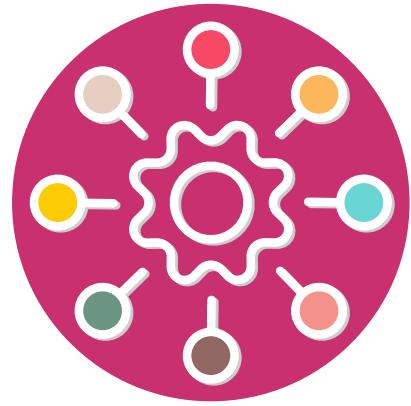


- Story written as a "User Story" (i.e. “As ... I want ... so that ...”)
- Acceptance criteria must exist and be understood by team
- User Story has been estimated by the team (in points)
- Where appropriate: Sketches, diagrams, annexes, reference documents and other resources exist, on a shared access platform
- The User Story is validated by the Product Owner
- The User Story is validated by the Dev Team

DoD – Definition of Done (Example)

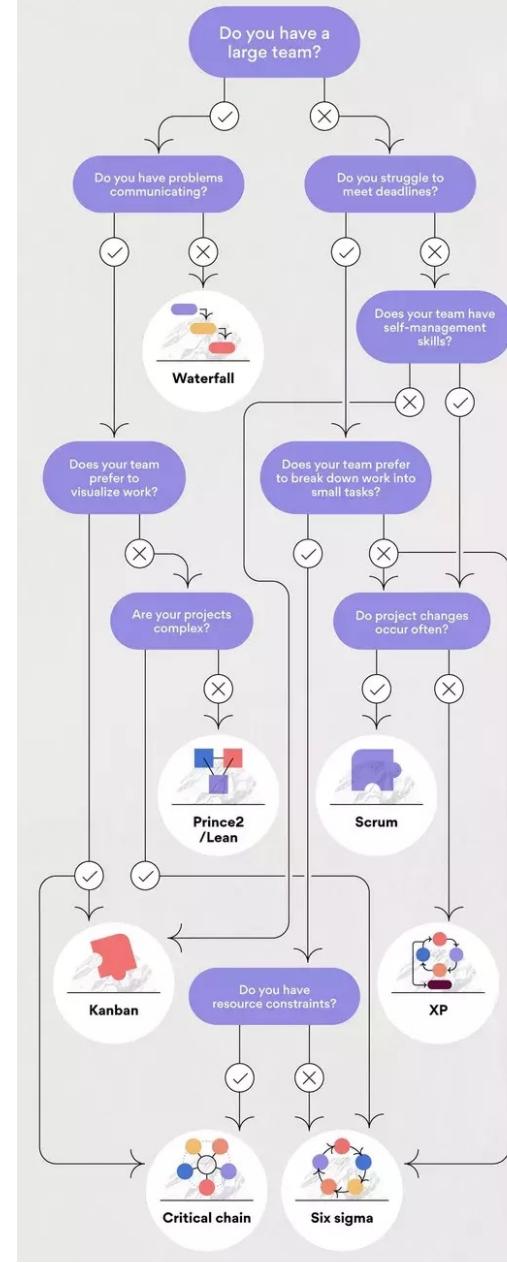


- If it is a code development User Story: The code builds and is archived on git
- If it is a code development User Story: The code has been reviewed and merged in a "master branch" on git
- If it is a code development User Story and if applicable: Some automated tests (checking for the User Story's Acceptance Criteria) are running on git
- If it is a code development User Story and if applicable : A documentation is available either as a ReadMe file in the repo or in the general documentation
- If the user story is not code related, All generated document are archived
- All the User Story's Acceptance Criteria are met and verified by the Team (e.g. in the Sprint Review)
- The User Story is accepted by the Product Owner



Outro

What is the right method ?



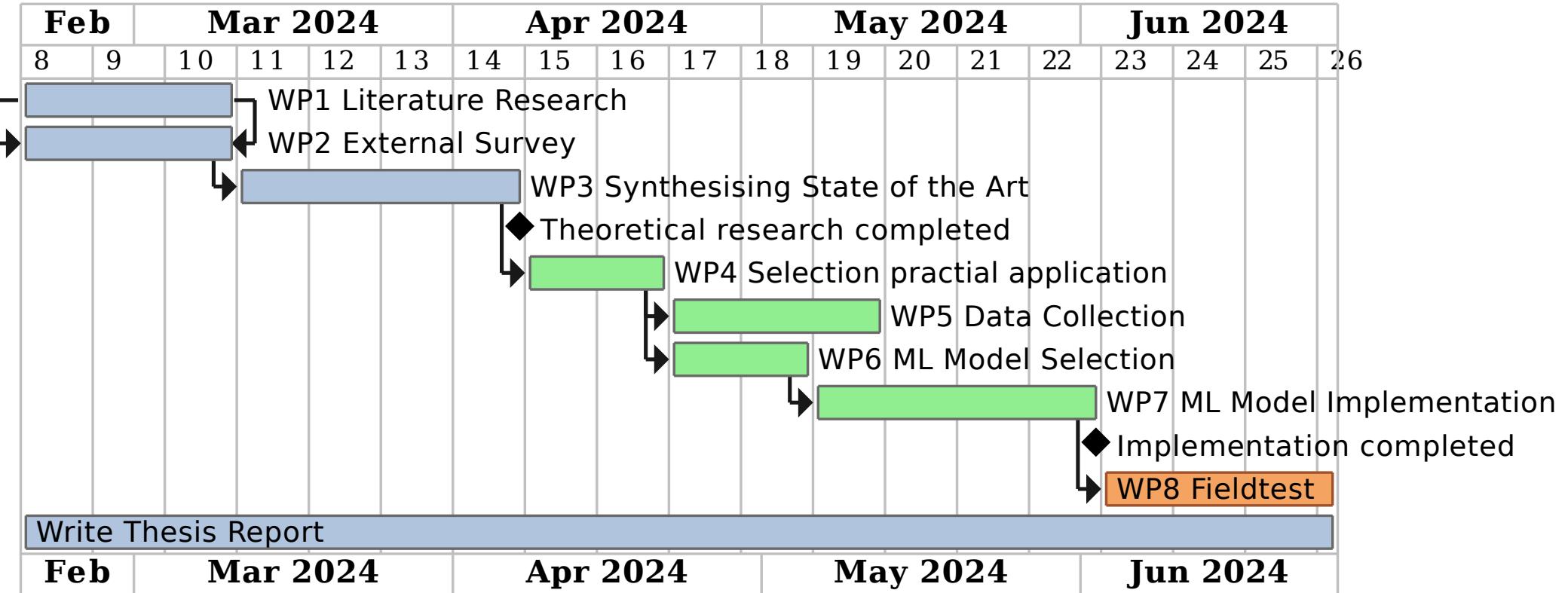
<https://asana.com/resources/project-management-methodologies>

ZaS

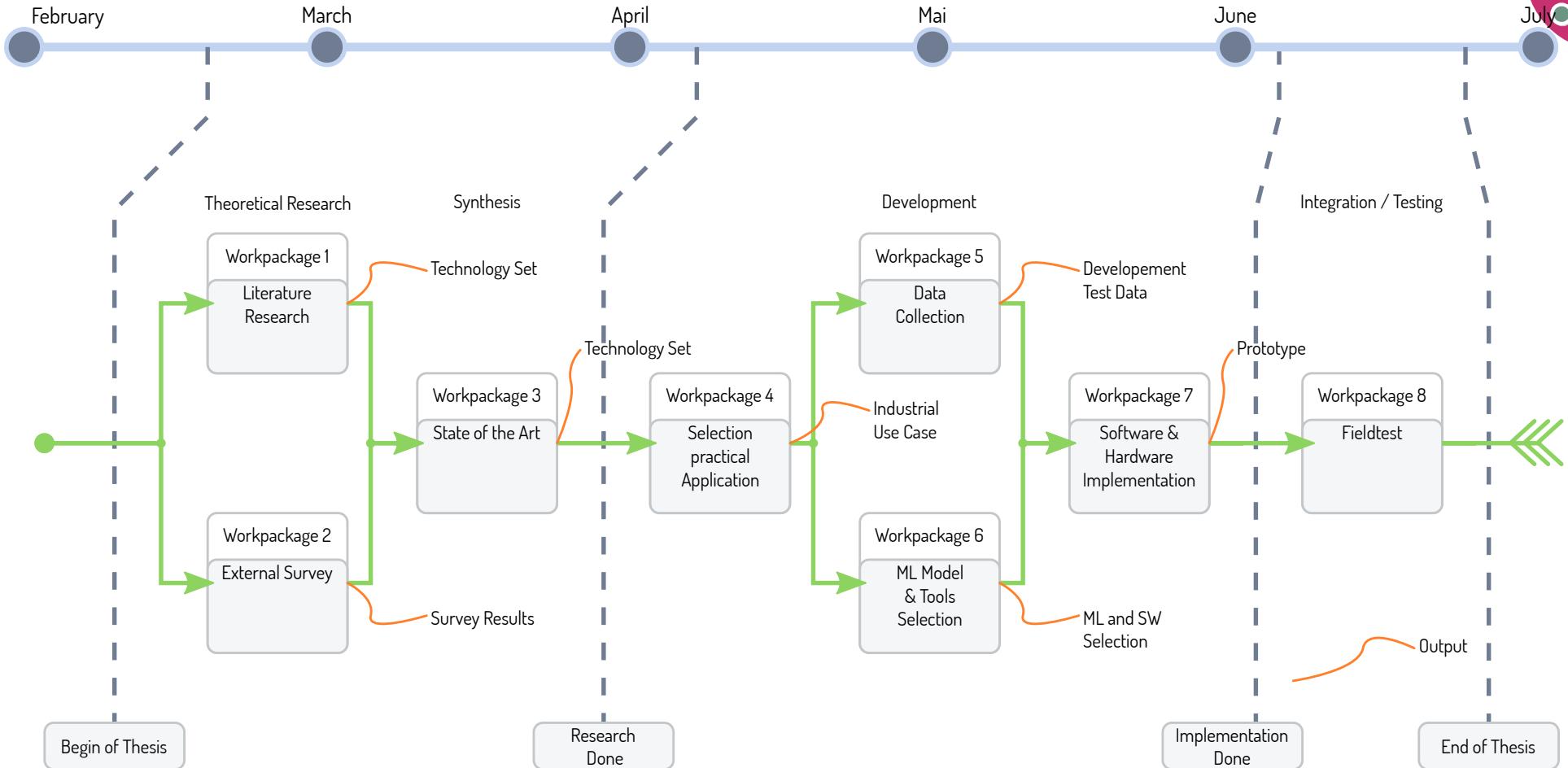
39



Gantt-Diagram



Project Plan



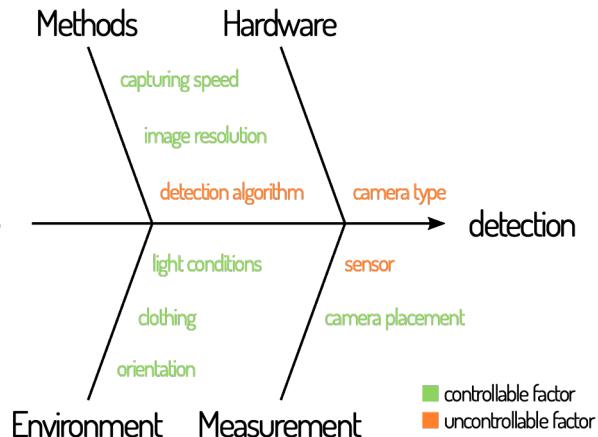
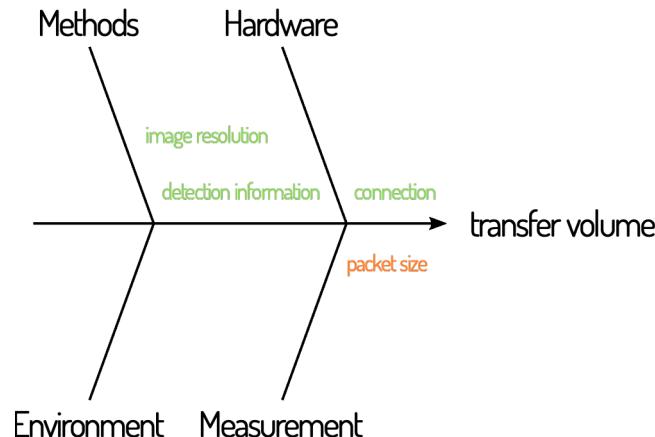
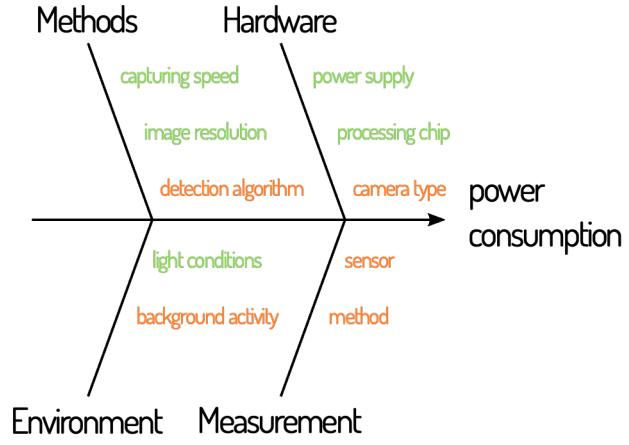
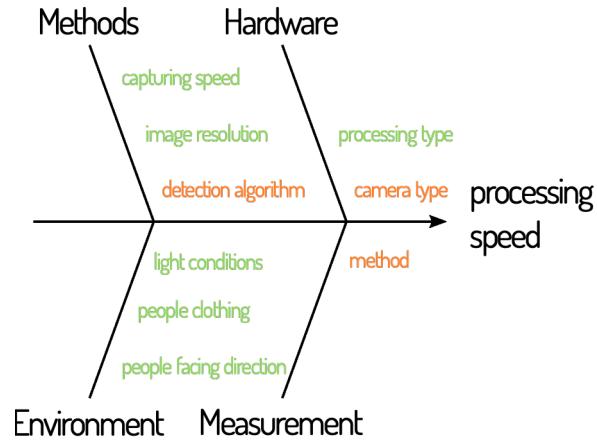
SWOT-Analysis



	Helpful for your objective	Harmful for your objective
Internal within technology	<h3>Strengths</h3> <ul style="list-style-type: none">• On-site processing, reducing bandwidth and latency• Longer hardware lifecycle• Realtime capabilities• Adaptive HW for future expansion• Reliable HW for security applications	<h3>Weaknesses</h3> <ul style="list-style-type: none">• Many ML nets are implemented and tested on other hardware first• Different programming and HW knowhow required• Custom hardware necessary depending on the application
External outside technology	<h3>Opportunities</h3> <ul style="list-style-type: none">• Other types of algorithms can be implemented• Developement of a new ecosystem• Proven technology also used in datacenters, sattelites and medical	<h3>Threats</h3> <ul style="list-style-type: none">• CPU and GPU technologies are getting faster and cheaper• User resistance to change to a new technology• Industial sector not used to• Smaller hardware choice available



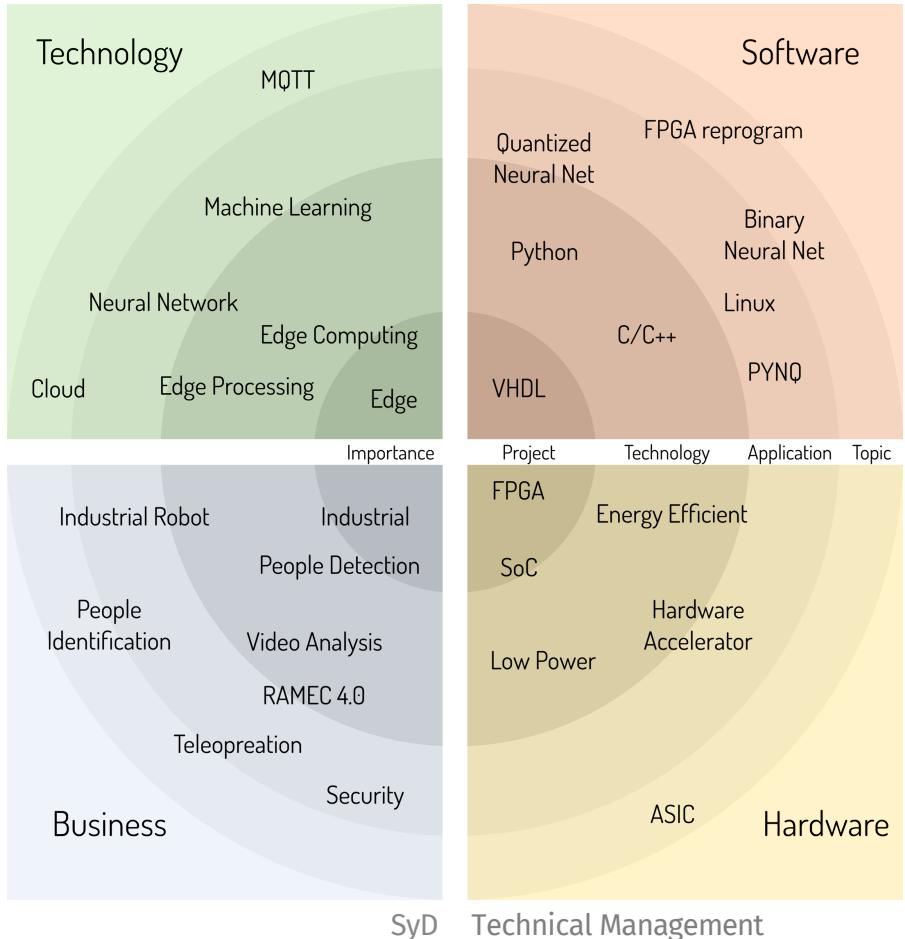
Cause and Effect Diagram

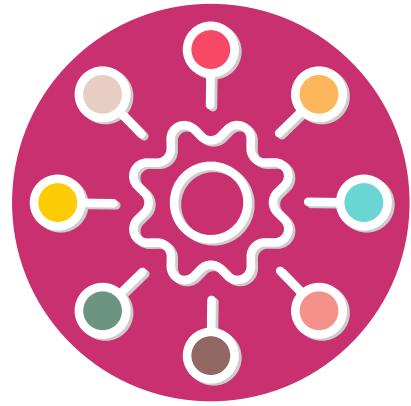


■ controllable factor
■ uncontrollable factor



Keyword Radar

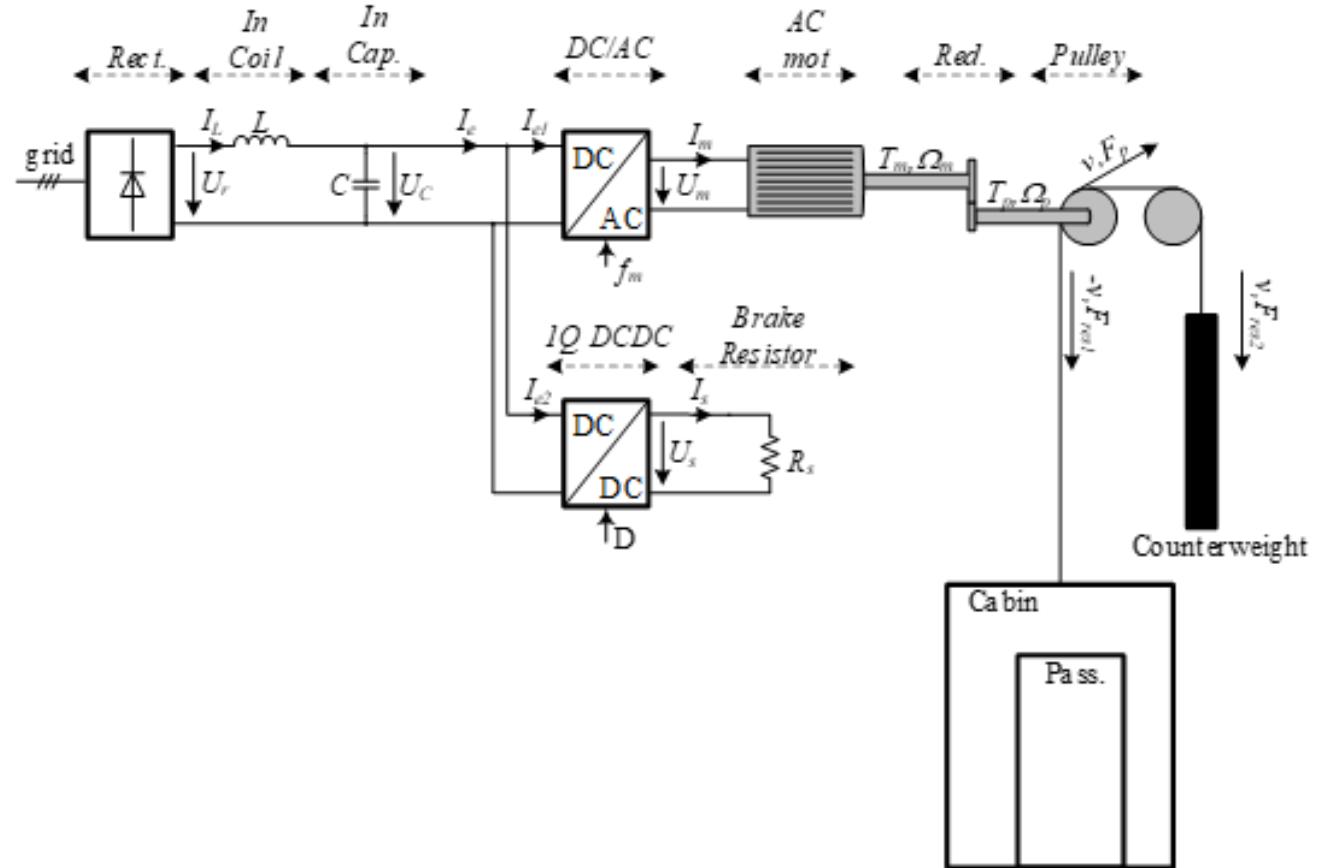


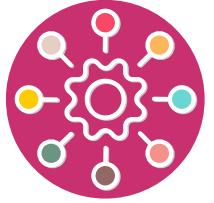


Example

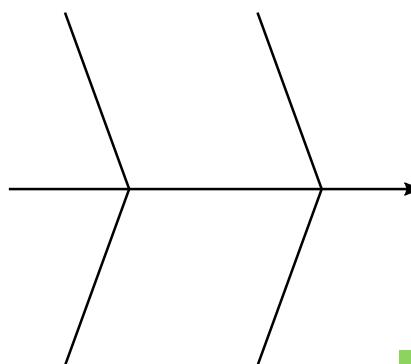
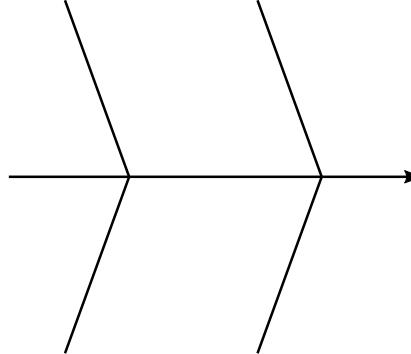
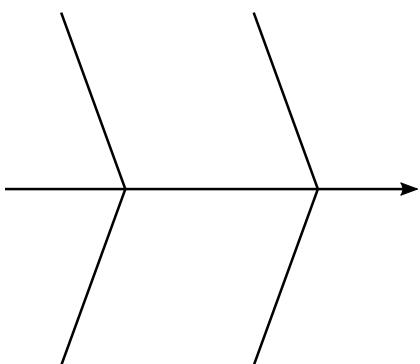
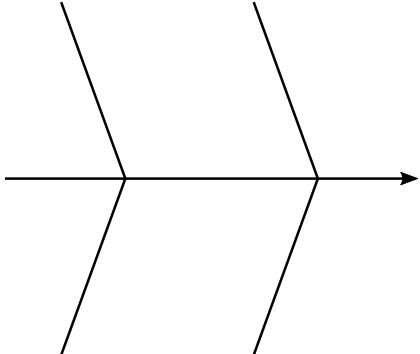


Generic scheme of the system "Elevator"





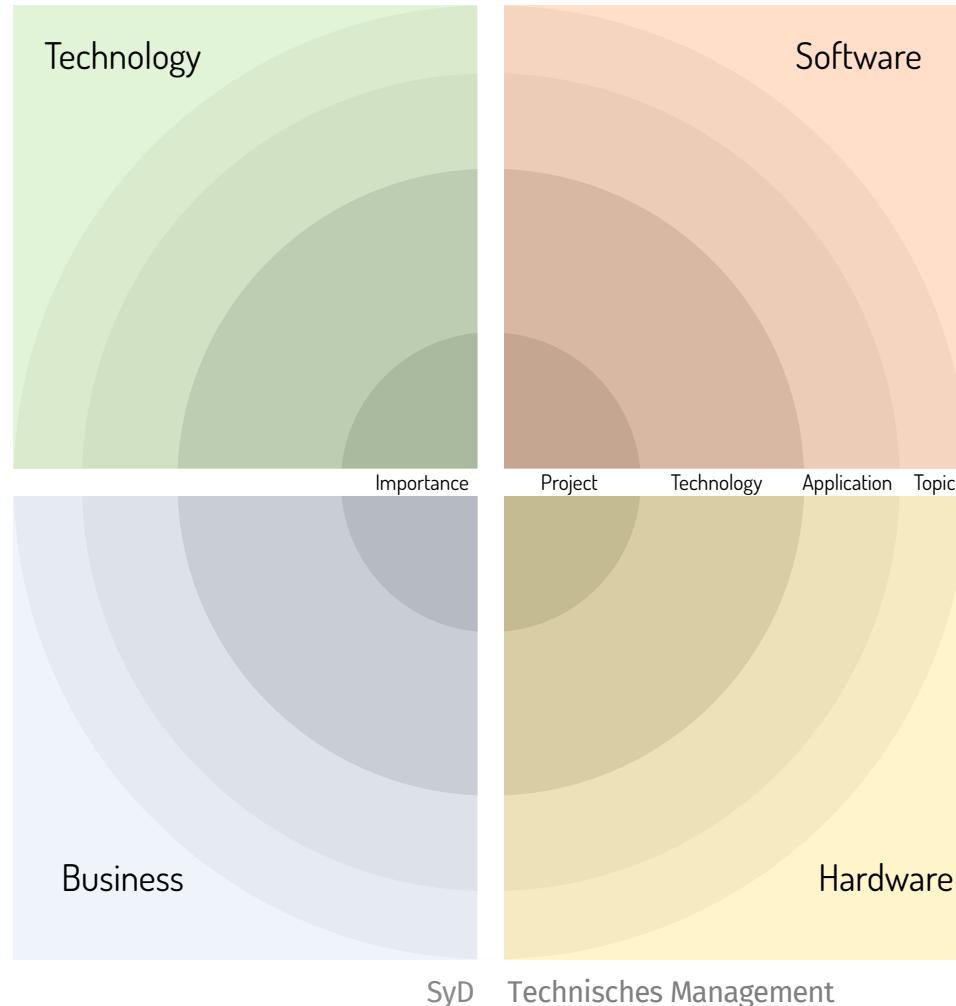
Cause and Effect Diagram



■ controllable factor
■ uncontrollable factor



Keyword Radar



UNCLASSIFIED



Per guidance from the Chief/DRRB CIA Declassification Center,
you may consider the document declassified... If you use an
exact copy of the document in your presentations, please draw
a line through the classification markings to prevent confusion.
Use the information as you see fit.

4/2/2008

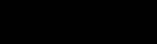
SIMPLE SABOTAGE
FIELD MANUAL

Strategic Services
(Provisional)

Sabotage Field Manual

STRATEGIC SERVICES FIELD MANUAL No. 3

UNCLASSIFIED



Project Interference



(11) General Interference with Organizations and Production

(a) Organizations and Conferences

(1) Insist on doing everything through "channels." Never permit short-cuts to be taken in order to expedite decisions.

(2) Make "speeches." Talk as frequently as possible and at great length. Illustrate your "points" by long anecdotes and accounts of personal experiences. Never hesitate to make a few appropriate "patriotic" comments.

(3) When possible, refer all matters to committees, for "further study and consideration." Attempt to make the committees as large as possible — never less than five.

(4) Bring up irrelevant issues as frequently as possible.

(5) Haggle over precise wordings of communications, minutes, resolutions.

(6) Refer back to matters decided upon at the last meeting and attempt to re-open the question of the advisability of that decision.

(7) Advocate "caution." Be "reasonable" and urge your fellow-conferees to be "reasonable" and avoid haste which might result in embarrassments or difficulties later on.

(8) Be worried about the propriety of any decision — raise the question of whether such action as is contemplated lies within the jurisdiction of the group or whether it might conflict with the policy of some higher echelon.

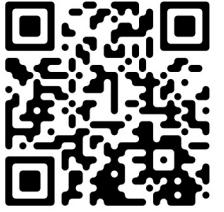


1. Open and direct communication
2. Be brief and precise
3. Direct line of communication
4. Wording is not important but understanding
5. Be fearless and bold

Task Estimation

Estimation 2

Create a small house
X Point(s)



0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	∞
?			

Task Estimation

Estimation 3

Create a programmable Lego robot

X Point(s)

