Engineering Software Product

Yudistira Asnar

yudis@staff.stei.itb.ac.id

Objective

- Students are aware the particularities of Software Engineering in Software Product Management
- Students understand various techniques commonly used in Engineering Software Product

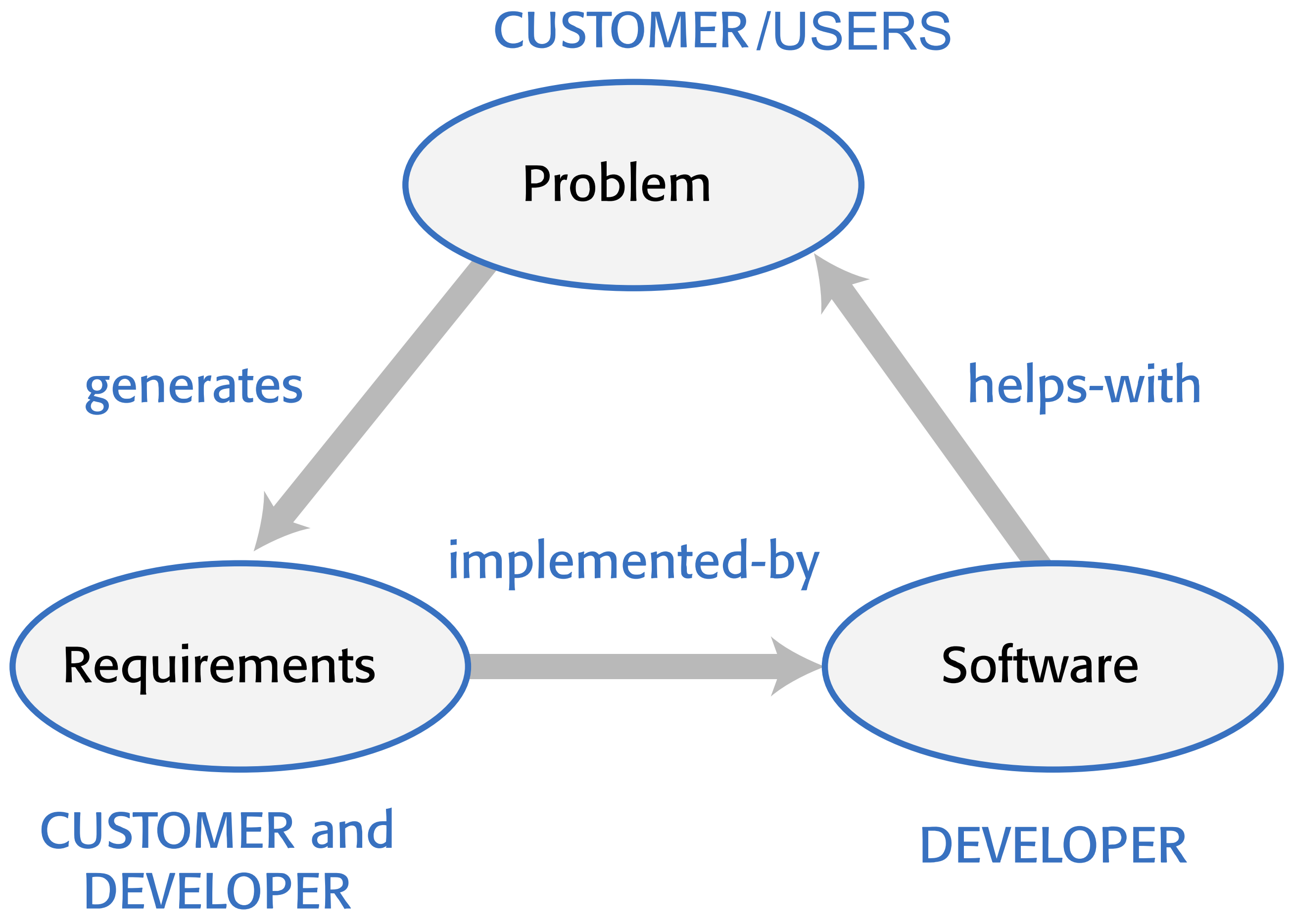
Reference

- Ian Sommerville, Engineering Software Product, 2018
- Eric Ries, The Lean Startup, 2011
- Ash Mauray, Running Lean: Iterate from Plan A to a Plan that Works, 2010

Software products

- Software products are generic software systems that provide functionality that is useful to a range of customers.
- Many different types of products are available from
  - large-scale business systems (e.g. MS Excel)
  - personal products (e.g. Evernote)
  - simple mobile phone apps and games (e.g. Suduko)
- Software product engineering methods and techniques have evolved from software engineering techniques that support the development of one-off, custom software systems.
- Custom software systems are still important for large businesses, government and public bodies. They are developed in dedicated software projects.

Project-based software engineering

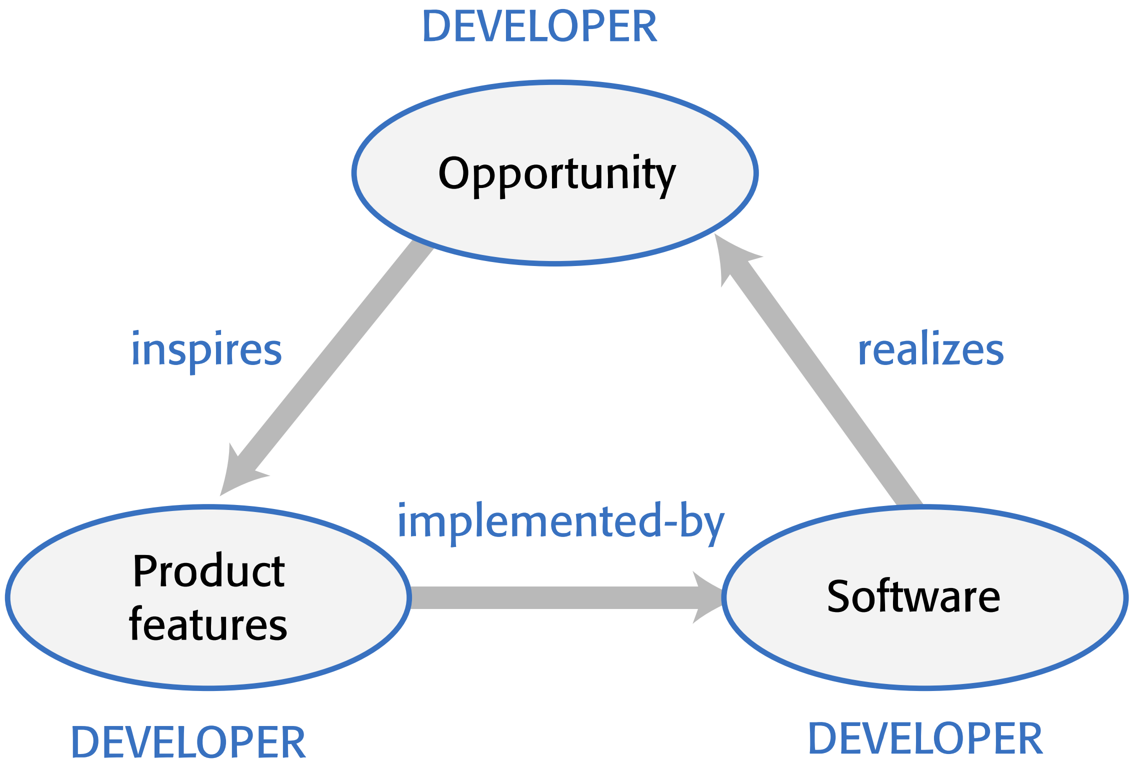


Adapted from Ian Sommerville 2018 (c)

Project-based software engineering

- The starting point for the software development is a set of ‘software requirements’ that are owned by an external client and which set out what they want a software system to do to support their business processes.
- The software is developed by a software company (the contractor) who design and construct a system that delivers functionalities to meet the requirements.
- The customer may change the requirements at any time in response to business changes (they usually do). The contractor must change the software to reflect these requirements changes.
- Custom software usually has a long-lifetime (10 years or more) and it must be supported over that lifetime.

Product software engineering



Adapted from Ian Sommerville 2018 (c)

Product software engineering

- The starting point for product development is a business opportunity that is identified by individuals or a company. They develop a software product to take advantage of this opportunity and sell this to customers.
- The company who identified the opportunity design and implement a set of software features that realize the opportunity and that will be useful to customers.
- The software development company are responsible for deciding on the development timescale, what features to include and when the product should change.
- Rapid delivery of software products is essential to capture the market for that type of product.
- Learning from market is essential to gain position in the market

Common Roles related to SW Product

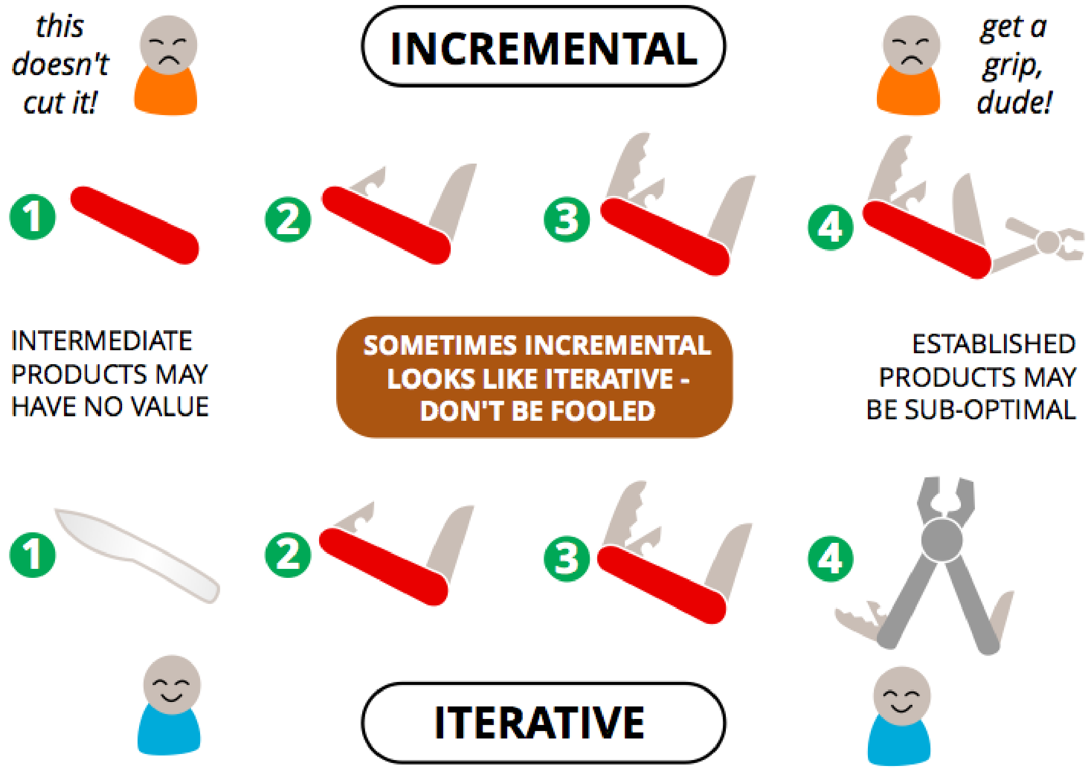
- Software Engineer
- Product Manager
- Designer

Common Practice

- Iterative approach is used to gain market and to learn from market
- Software is delivered in rapid fashion
- Software collects various analytics to learn from users (market) and act accordingly (agile)
- Software has various version and variation
  - Version: alpha, release candidate, 1.0, 2.1, 3.1.3
  - Variation: android mobile, android autos, android things
  - Shares some codes (code base, platform, core library)

Iterative vs Incremental

- In each cycle, customers use the software and give some feedback
  - What is actually work
  - What is actually needed
  - What is actually being used



this doesn't cut it!
INCREMENTAL
get a grip, dude!
1
2
3
4
INTERMEDIATE PRODUCTS MAY HAVE NO VALUE
SOMETIMES INCREMENTAL LOOKS LIKE ITERATIVE - DON'T BE FOOLED
ESTABLISHED PRODUCTS MAY BE SUB-OPTIMAL
1
2
3
4
ITERATIVE

https://safetydave.net/iterative-vs-incremental/

Lean Software Development

Origin: Mary Poppendieck, Tom Poppendieck, Thomas David Poppendieck on Lean Software Development

Principles

- Eliminate waste
- Amplify learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity
- See the bigger picture (Optimize the whole)

“The big question of our time is not Can it be built? but **Should it be built**? This places us in an unusual historical moment: our future prosperity depends on the quality of our collective imaginations.”

Eric Ries in The Lean Startup

An Example

- Iridum 9500 (Sat Phone) – 1987 - 1998

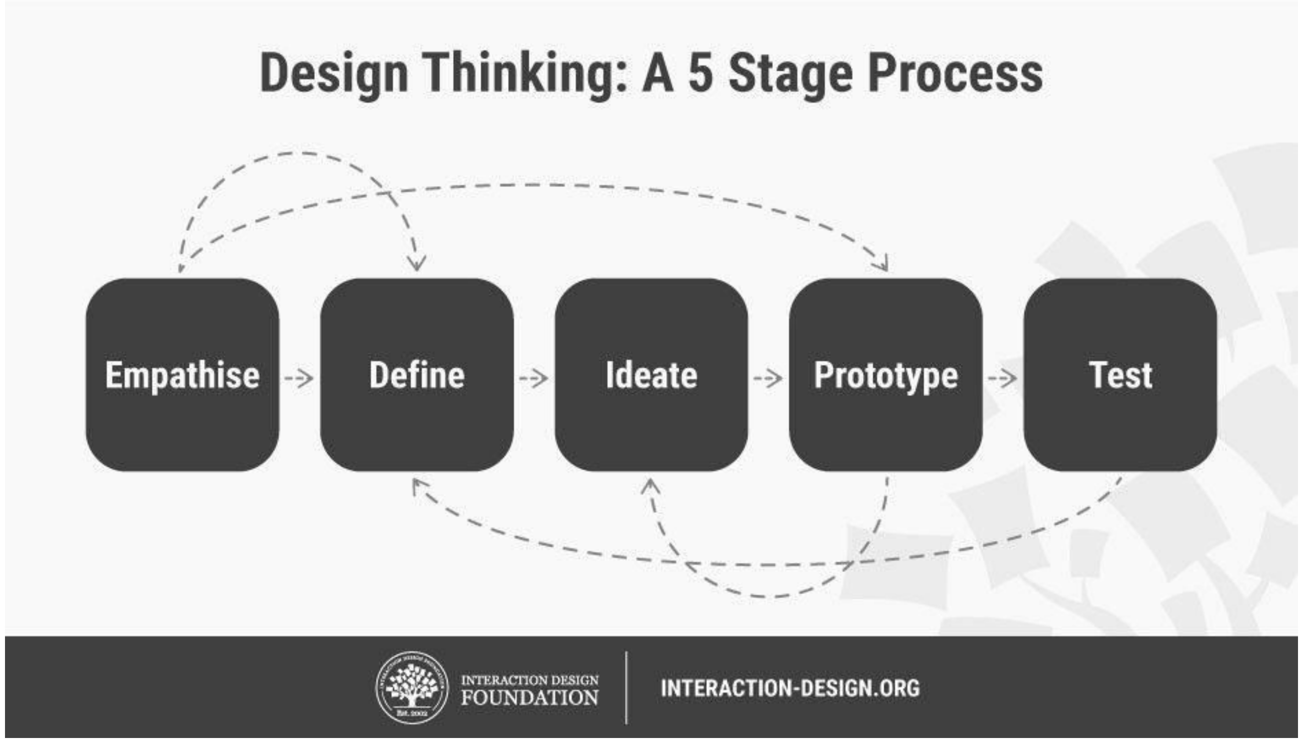
https://steveblank.com/2010/11/01/no-business-plan-survives-first-contact-with-a-customer-–-the-5-2-billion-dollar-mistake/

Some main reasons failed products

- We love our solution, thus… We build something NOBODY WANTS ….
- Product never get started; too much time analyzing, planning, and thinking…
- Run out cash/budget
- Too late to enter the market
- Unsuited to the market
- Improper business model
- Legal challenges
- Burn out, lack of talent

Design Thinking

- Empathize
- Define
- Ideate
- Prototype
- Test

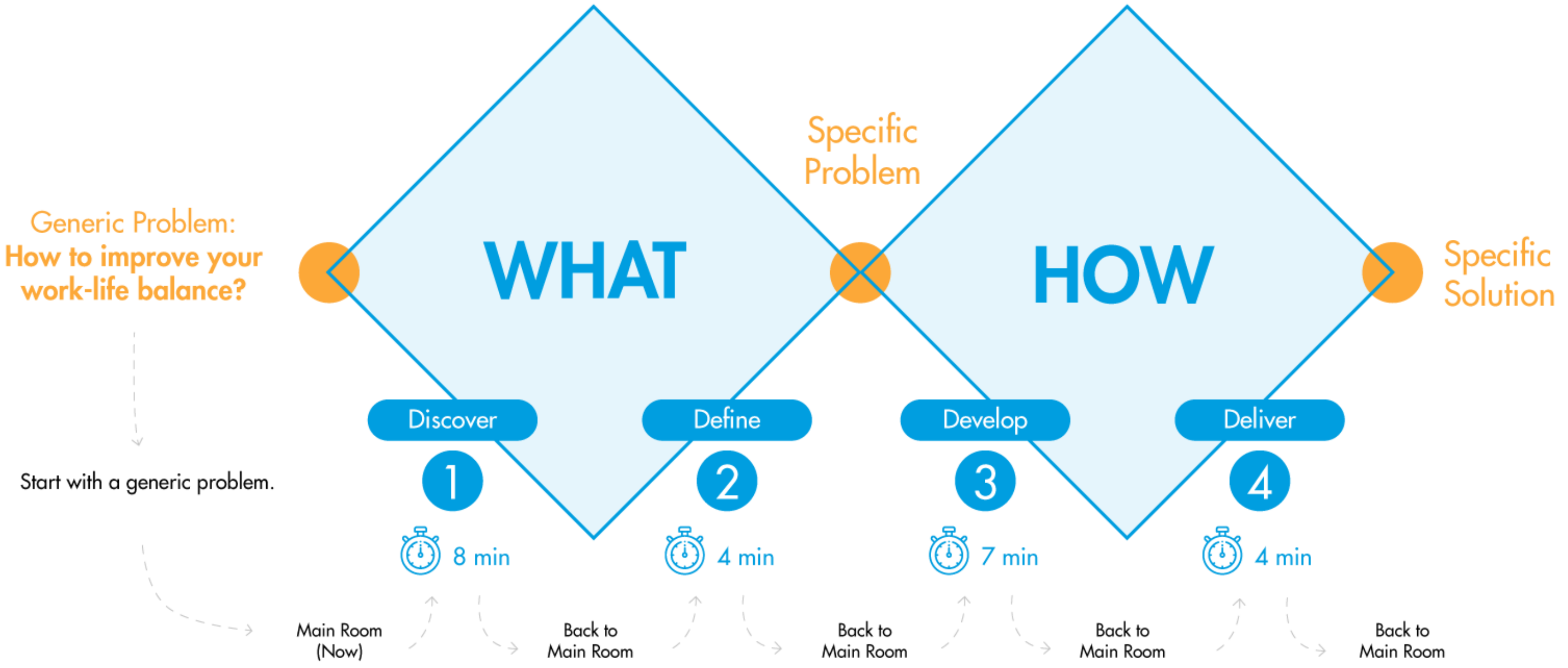

Design Thinking: A 5 Stage Process
Empathise
Define
Ideate
Prototype
Test
INTERACTION DESIGN FOUNDATION
INTERACTION-DESIGN.ORG

Balance

Ideation, Inspiration, Implementation

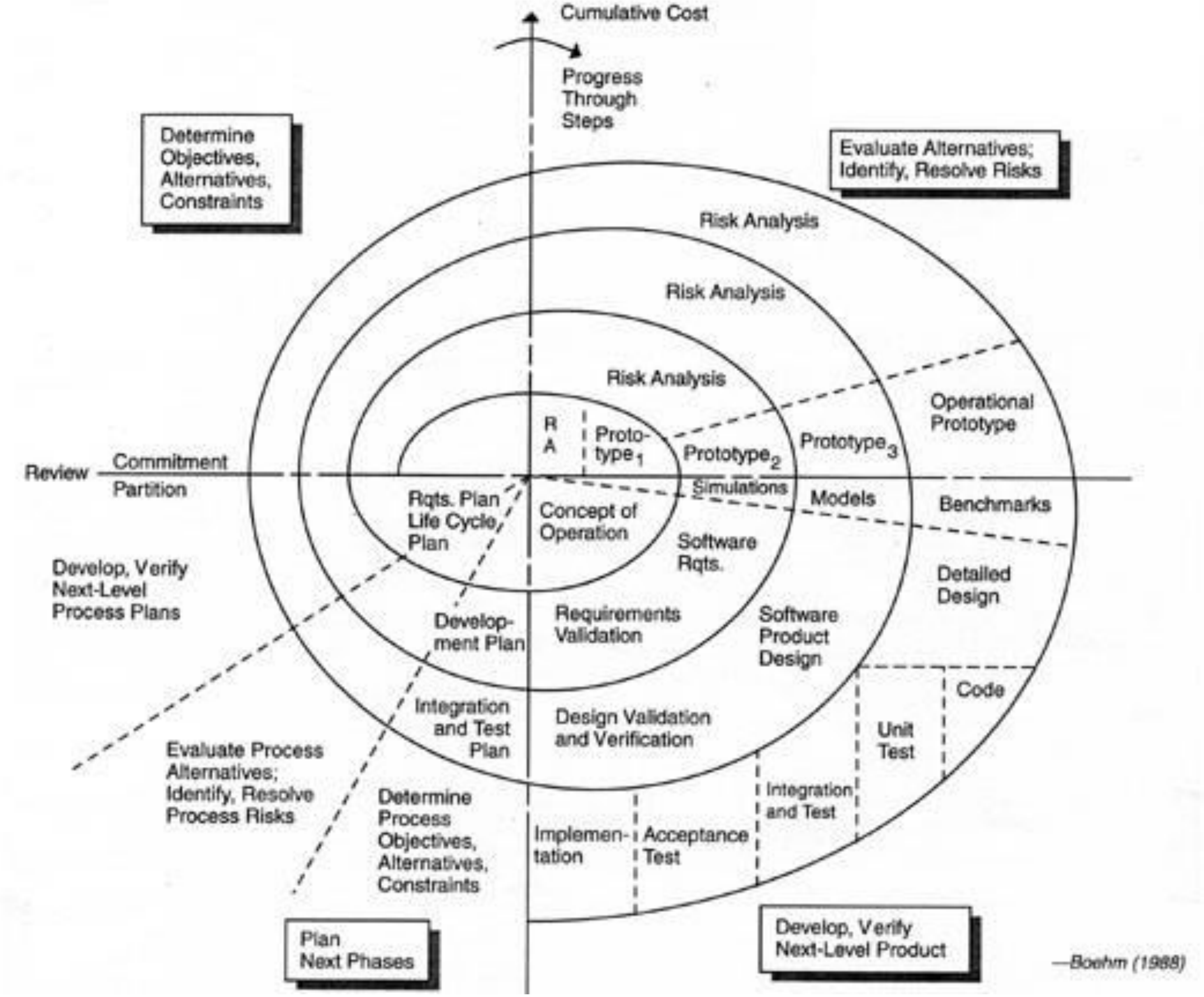
Design Thinking Workshop

- Double diamond
  - Diverge
  - Converge



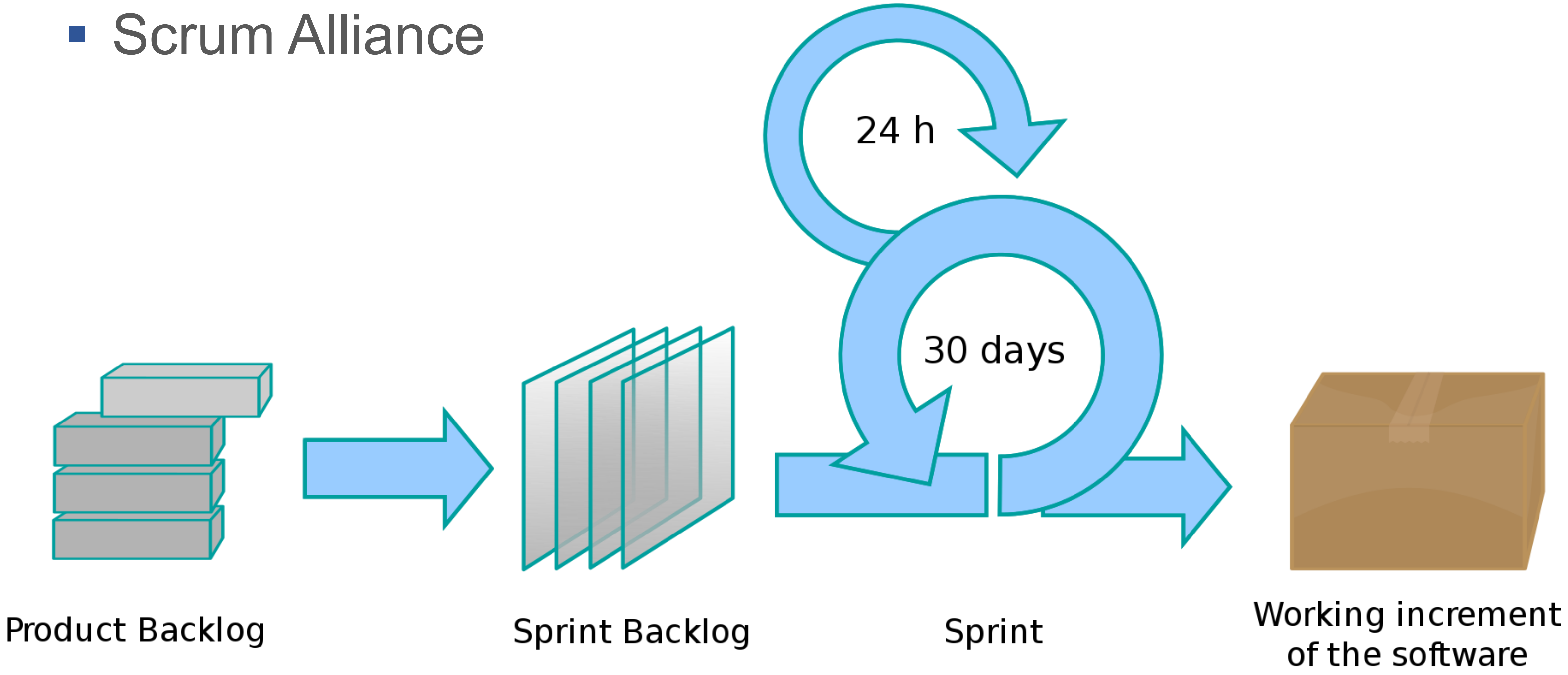
Some resemblance with Spiral Model

- Boehm, 1988



Some resemblance with Scrum

- Scrum Alliance



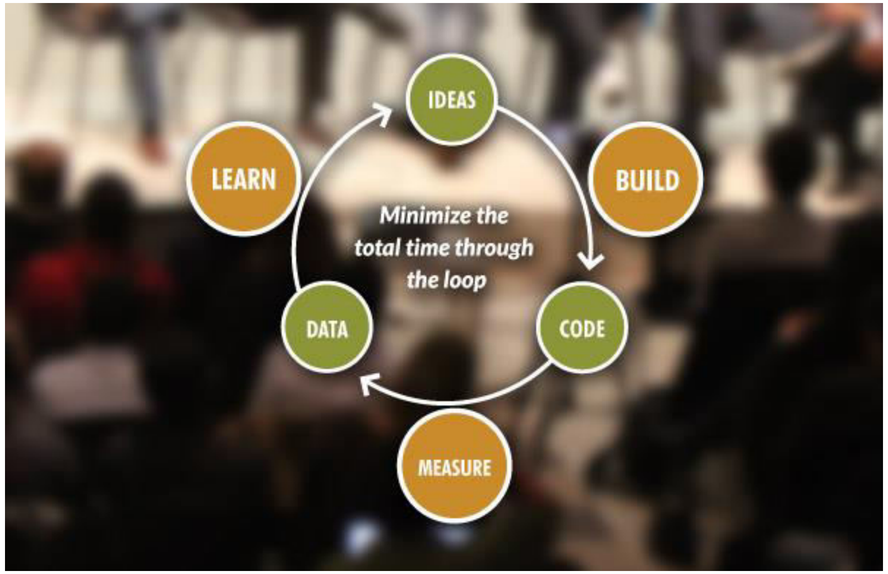
wikipedia.org

Lean Startup

Origin: Eric Ries on The Lean Startup

Principles

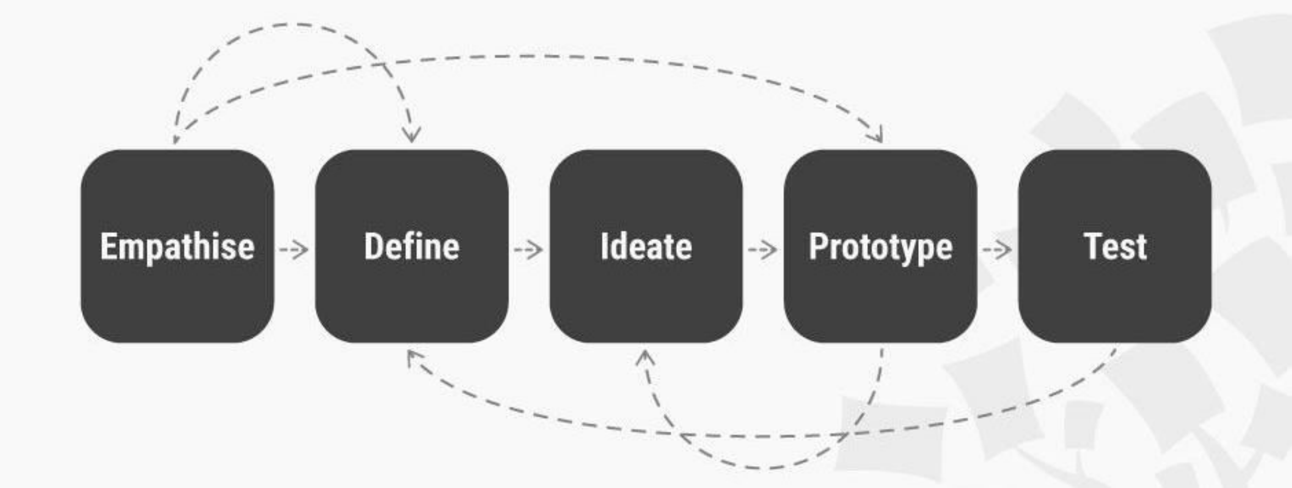
- Minimum Viable Product
- Continuous deployment
- “Test” Frequently and Lean Quickly
- Actionable metrics
- Pivot based on key learnings
- Innovation accounting

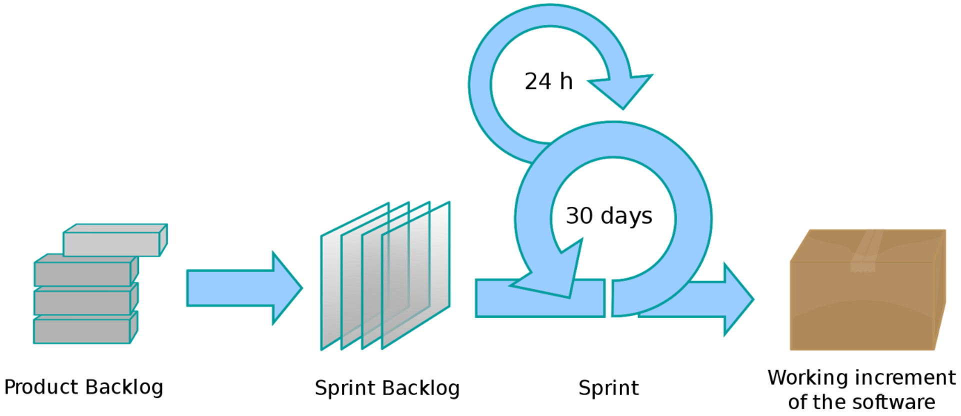


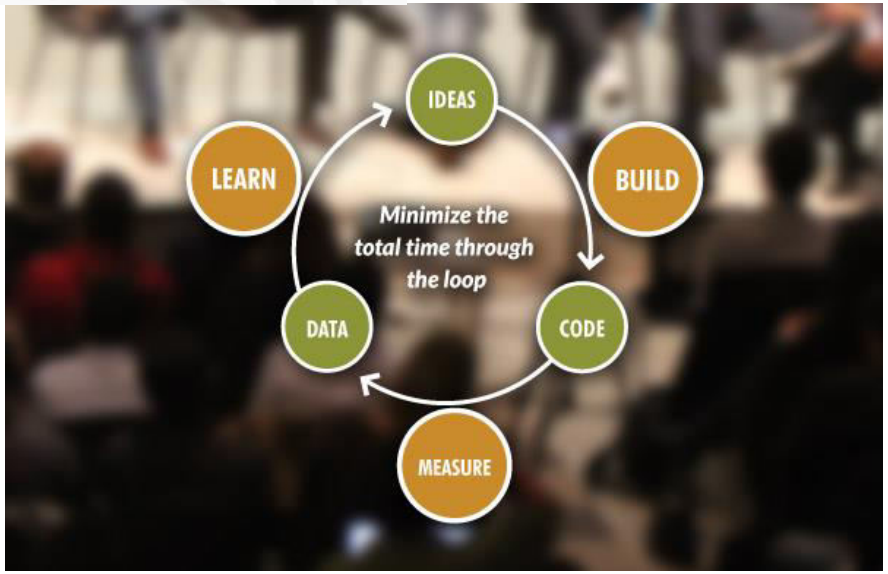
http://theleanstartup.com/

Build-Measure-Learn

Interplay


Empathise
Define
Ideate
Prototype
Test

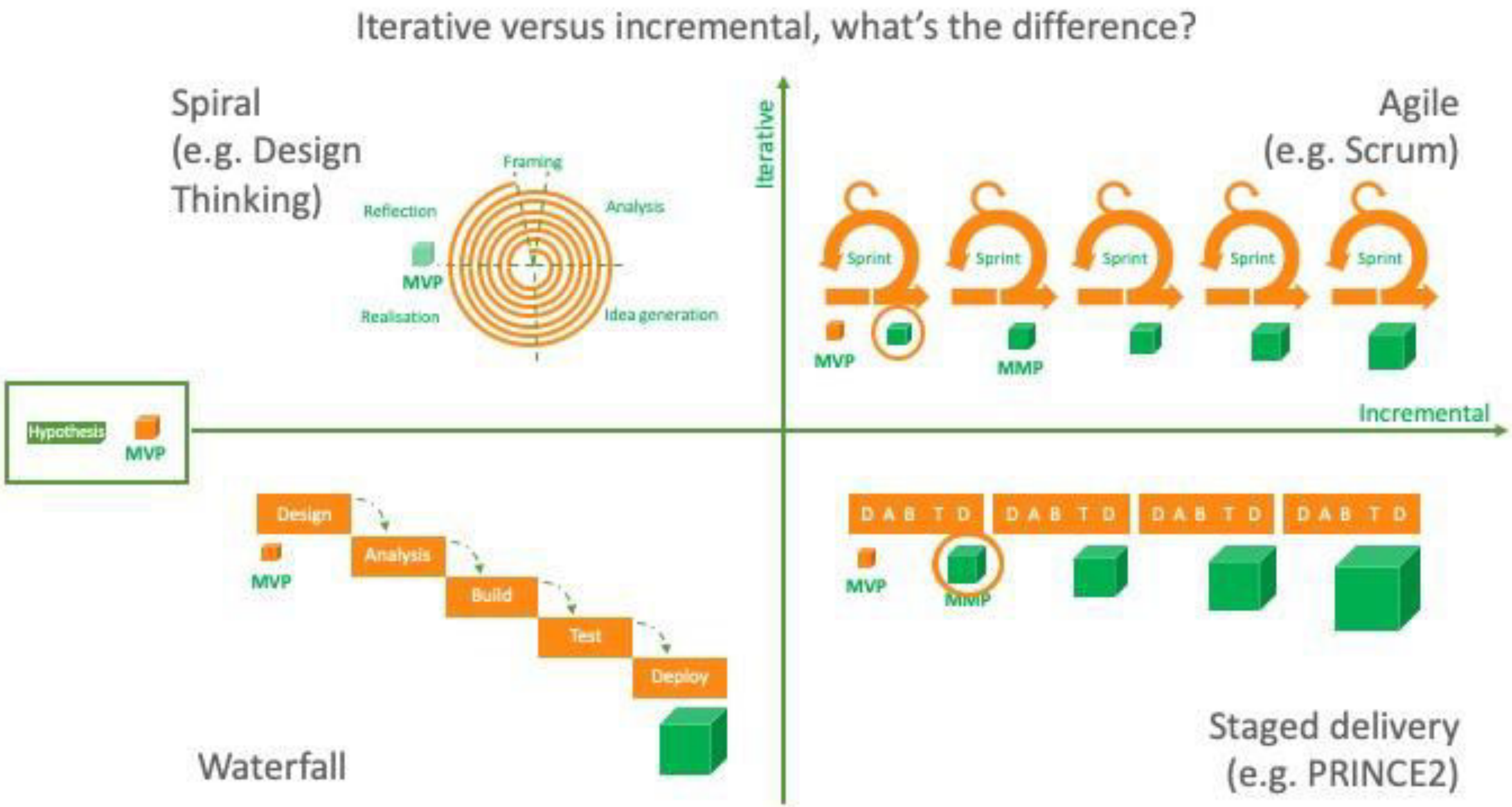

24 h
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment of the software


IDEAS
BUILD
CODE
MEASURE
DATA
LEARN
Minimize the total time through the loop

Common Principles

- Customer value == business value
- Work in short cylces
- Hold regular retrospective
- Go & See
- Test high risk hypothesis
- Do less more often
- Work as a balanced team
- Transparency
- Review incentive structures
- Make learning as 1st priorties

Incremental – Iterative Dimension



https://hennyportman.wordpress.com/2020/03/20/are-incremental-and-iterative-the-same-phenomenon-or-not-part-2/

Reference

- Ian Sommerville, Engineering Software Product, 2018
- Eric Ries, The Lean Startup, 2011
- Ash Mauray, Running Lean: Iterate from Plan A to a Plan that Works, 2010