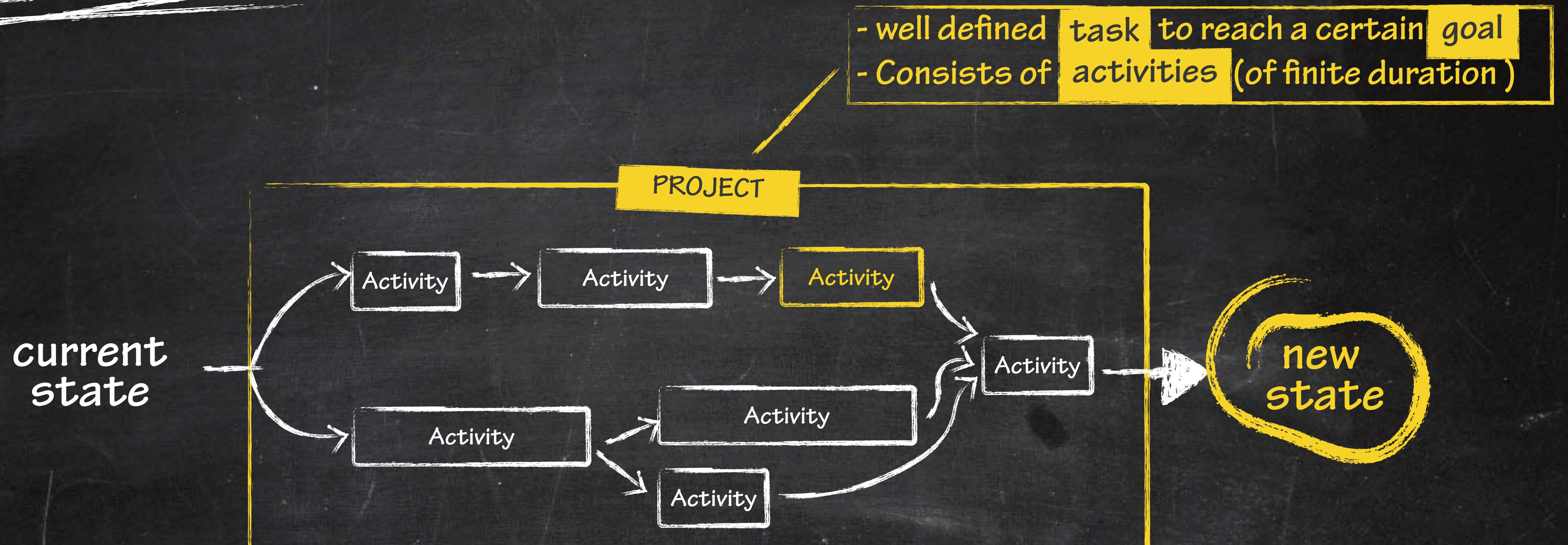


**PROJECT
MANAGEMENT &
SOFTWARE
DEVELOPMENT**

What's a Software-Project and what's its Management?

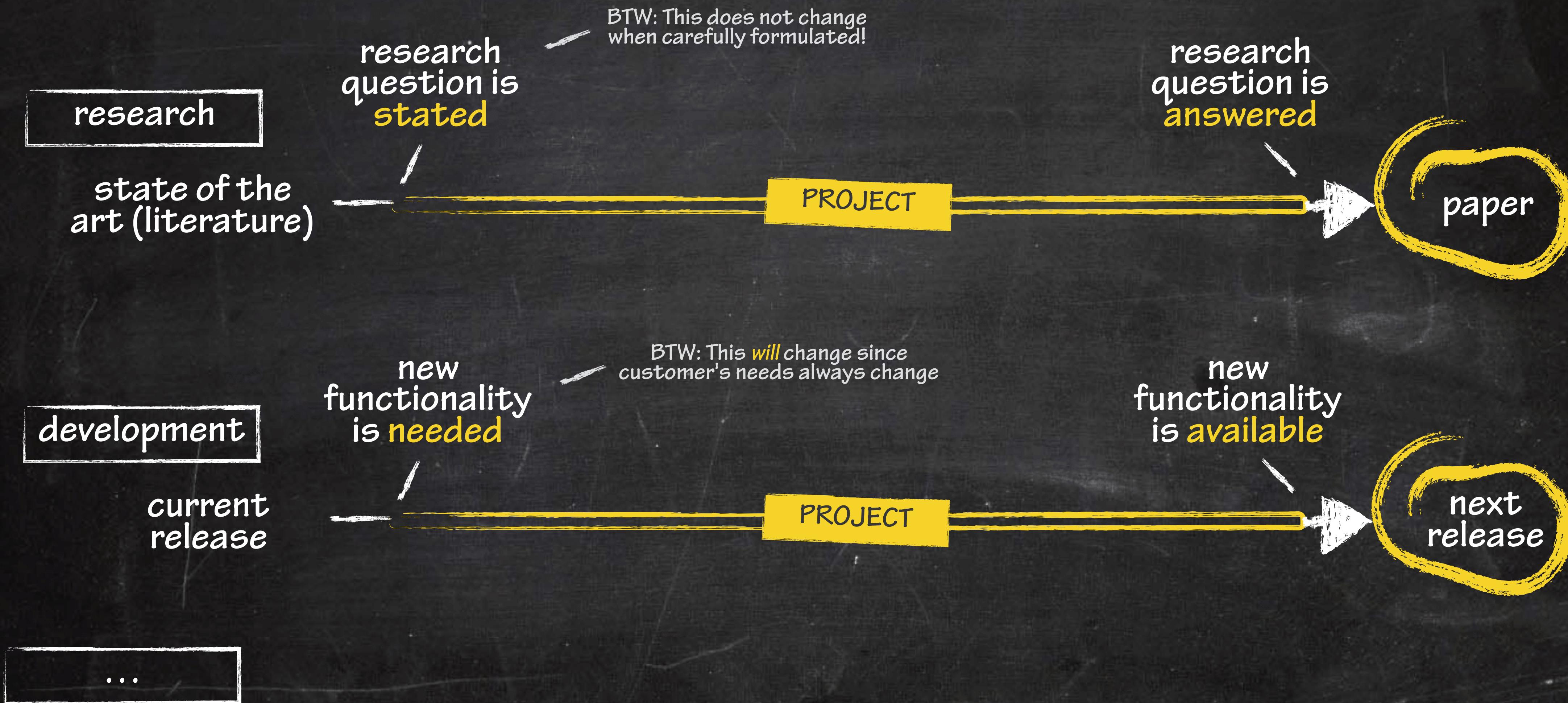


PROJECT
MANAGEMENT

the discipline of planning, implementing, monitoring and controlling a software project (i.e., defining and reaching targets while optimizing the utilization of resources)

What's a Software-Project and what's its Management?

Examples



Reasons for Failed Projects

changing requirements
goal-setting was unrealistic

falsely estimated durations

product quality is not good enough

incompetent workers

client was not involved enough

not well-defined requirements

deadline exceeded

not enough resources

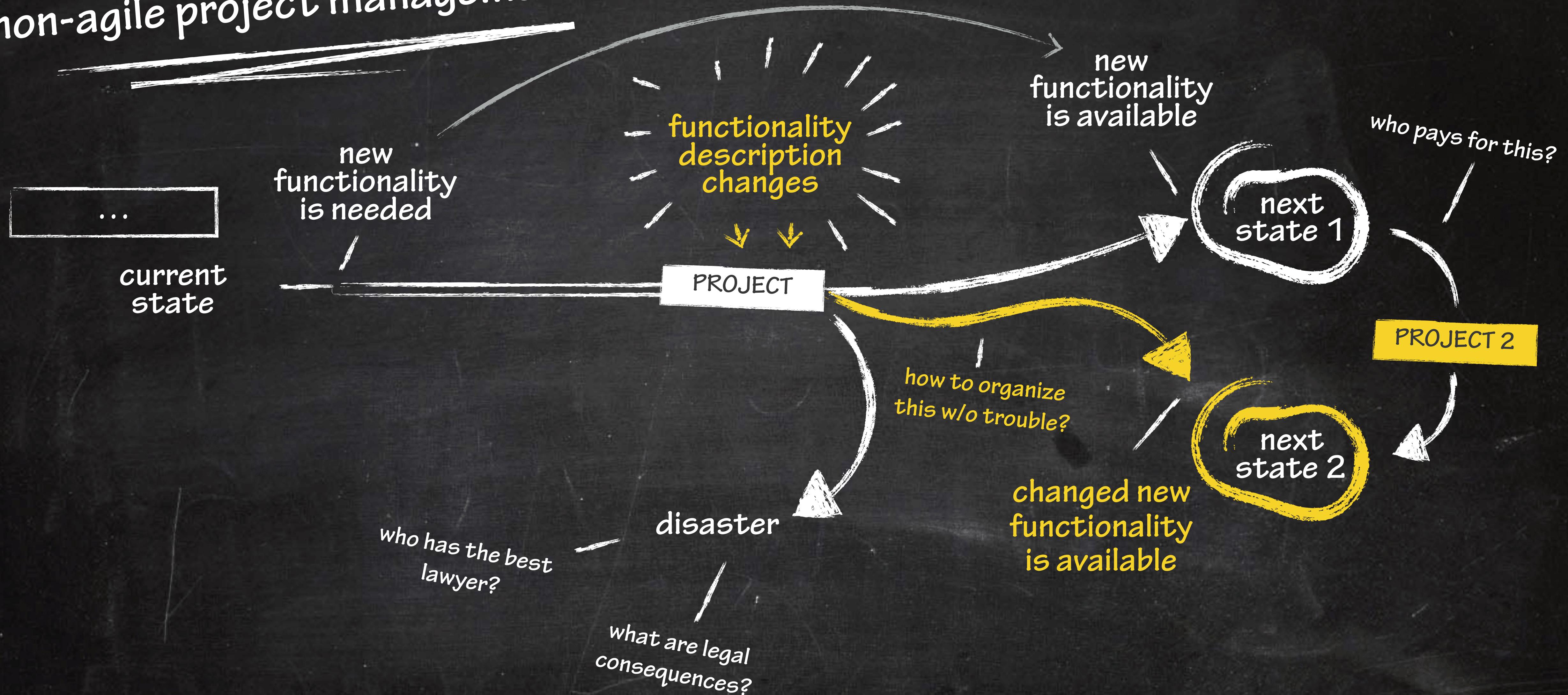
falsely estimated costs

no market for the product

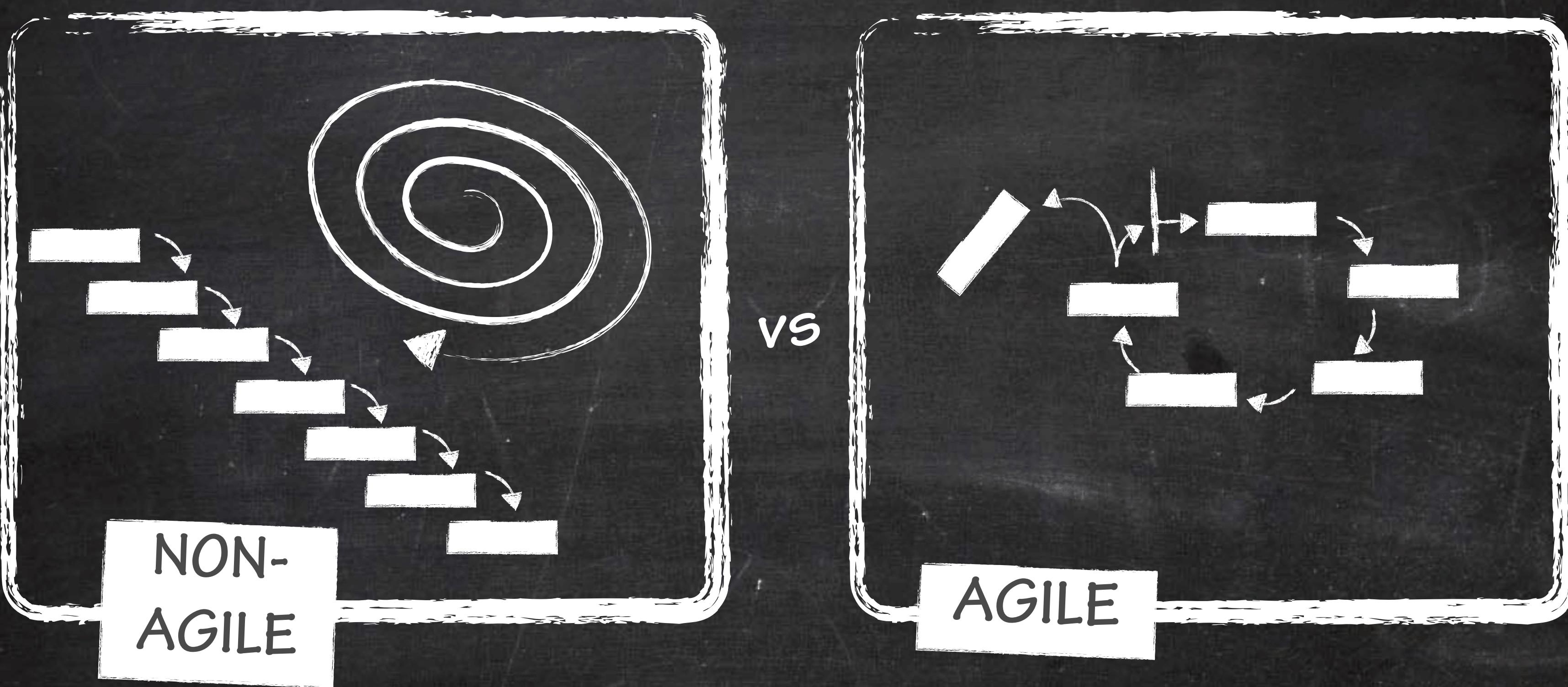
try to minimize these risks

PROJECT

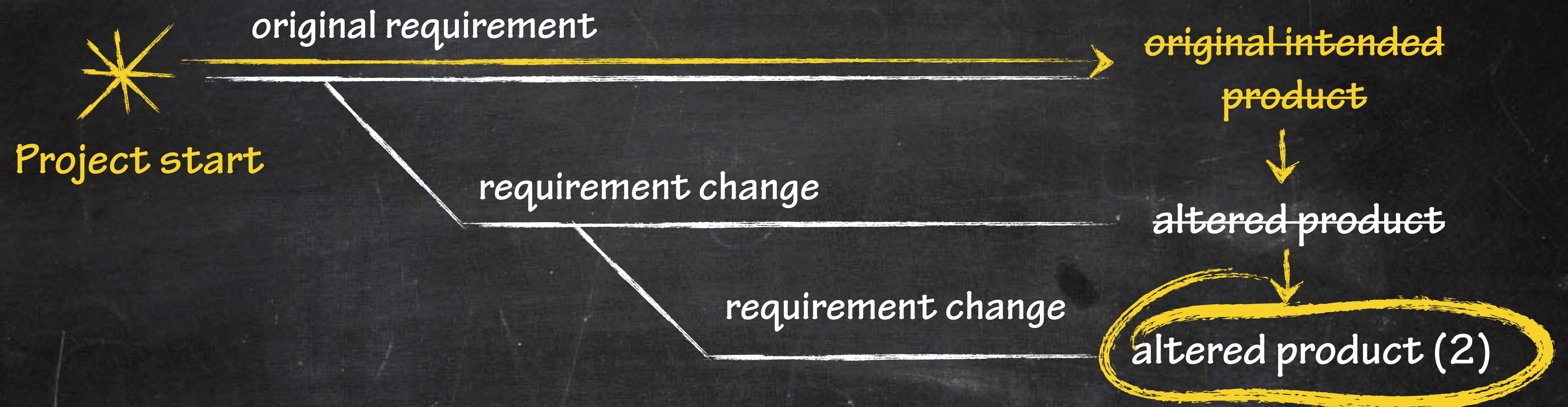
Changes in requirements for non-agile project management



Development Methodologies

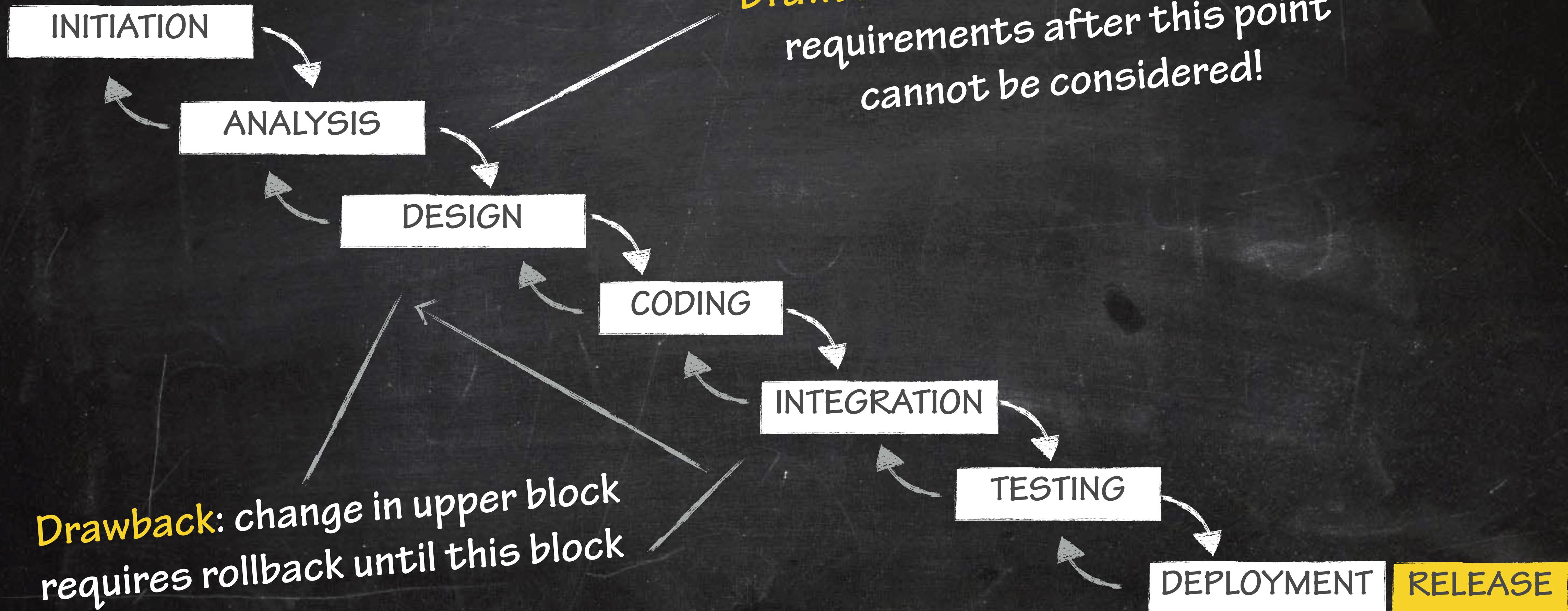


Non-Agile Methodologies



The Waterfall Model (1968)

→ regular
← extended



The Spiral Model

determine objectives

plan next iteration

cumulative costs

identify and resolve risks

reviews

development and test

Release

analysis plan

INITIATION

concepts for analysis and operations

development plan

test plan

Prototype 1

Prototype 2

Operational Prototype

draft

DESIGN

CODE

INTEGRATION

TEST

DEPLOYMENT

Release

analysis plan

INITIATION

concepts for analysis and operations

development plan

test plan

Prototype 1

Prototype 2

Operational Prototype

draft

DESIGN

CODE

INTEGRATION

TEST

DEPLOYMENT

Release

Manifesto for Agile Software Development

*Individuals and interactions
over processes and tools*

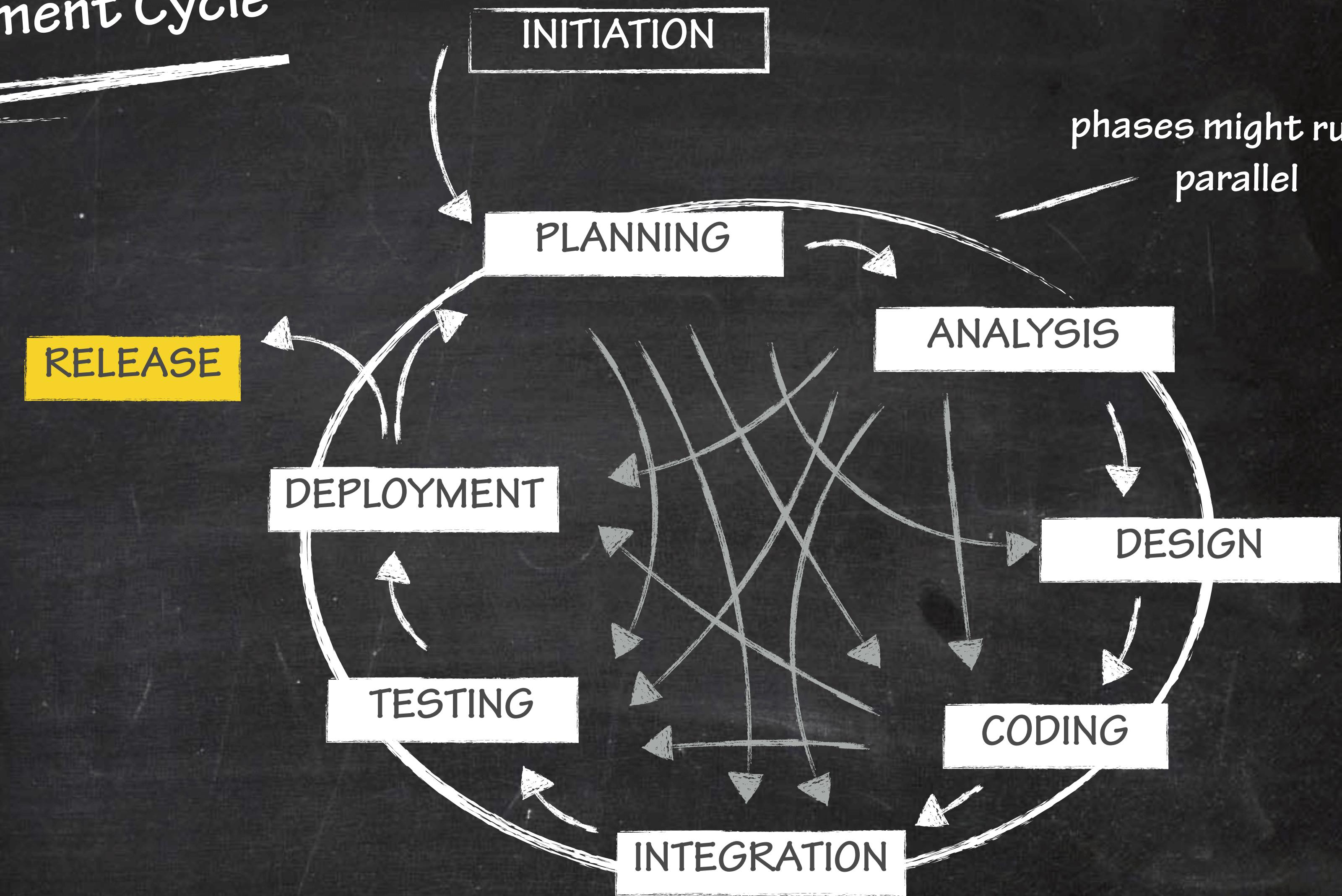
*Working software
over comprehensive documentation*

*Customer collaboration
over contract negotiation*

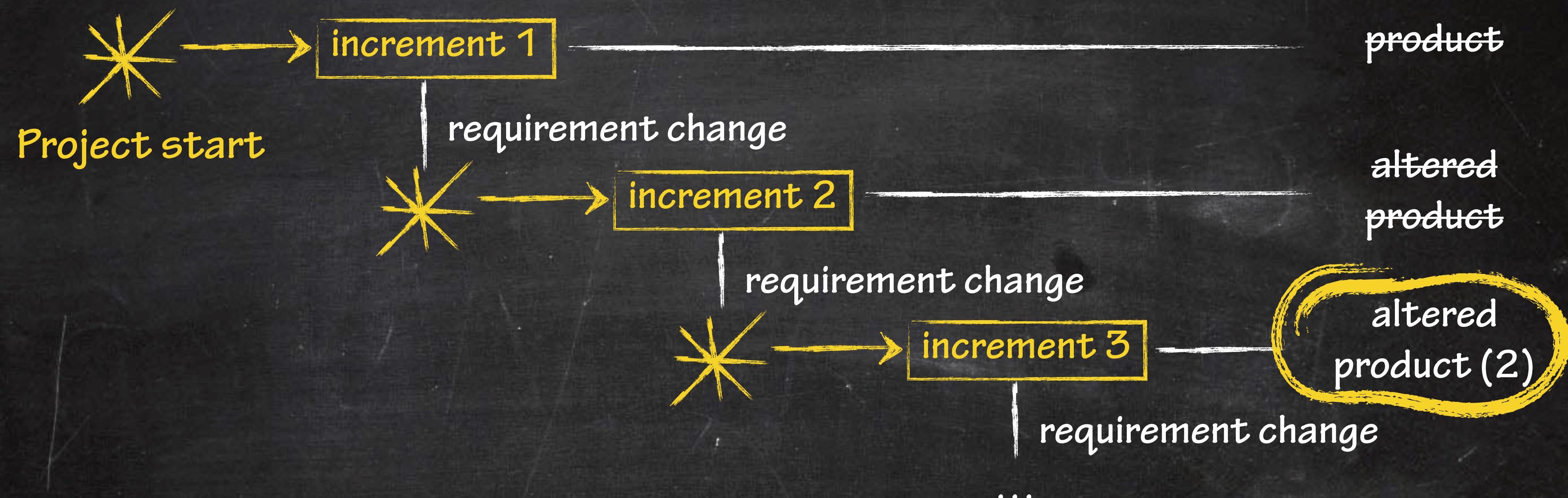
*Responding to change
over following a plan*

*Agile software
development is an idea
not a concrete method.*

The Agile Development Cycle



Agile Methodologies



Agile Methodologies

Characteristics

- focus on **flexibility**, **continuous improvements**, and **early releases**
- **incremental** and **iterative**
- **periodic & continuous communication** (leaders, team, stakeholders, customers)
- **open to changes** in requirements
- **emphasizes feedback** from customers / end users
- **mixed team** works on each product iteration
- **prioritizing of work units** based on value (business or customer)
- each **iteration** results in a **working product**

Agile Methodologies

Pro's & Cons's

PROs

- **Change is first class citizen:** short planning cycles, acceptance for changes at any time in the project
- **Open-end is okay:** a final goal must *not* be known in advance, it's becoming clear during dev-time
- **Early high-quality builds:** project is sequence of iterations where each one is tested and carefully developed
- **Focus on communication:** team work is important; team mates communicate face-to-face & own project parts + share and advance their skills together

CONS

- **Sloppy documents :** no reliable estimation on final delivery object, patch-work documentation and documents
- **Changing plans:** work items scoped to time-boxed iterations; item move implies importance gain or loss; hard to guarantee in advance that work item is contained in certain iteration
- **Need for higher expertise :** teams need highly skilled developers (on multiple topics) who are bound to project

Concrete Methods for Agile Software Development

focus: improve quality and responsiveness to evolving customer requirements

Extreme Programming (XP)

principles: include feedback, assuming simplicity, embracing change

focus: visualize things for small, continuous changes

Kanban

5 principles: workflow visualization, work in progress limitation, flow management and enhancement, explicit policies, continuous improvements.

focus: iteratively follow a set of roles, responsibilities, and fixed meetings

Scrum

most popular

principle: plan, schedule, develop and deliver increments each week („sprint“), deliver regularly

focus: iterative and incremental based on industry best practice

Feature-Driven Development (FDD)

focus: remove work that does not add value to a product or service

Lean Software Development (LSD)

7 principles: eliminate waste, amplify learning, favor late decisions, early delivery, team empowerment, build integrity in, see the „big picture“

Based on: <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

focus: avoid typical failures in projects (budget exceeded, missed deadline,...)

Dynamic Systems Development Method (DSDM)

8 principles: focus on business need, deliver on time, collaborate, ensure quality, build incrementally and iteratively, continuous clear communication continuously, control

focus: projects should continuous adapt

Adaptive System Development (ASD)

cycle of 3 activities: speculate, collaborate, learn



SOFTWARE
DESIGN

SOFTWARE
DEVELOPMENT

MANAGEMENT & COLLABORATION