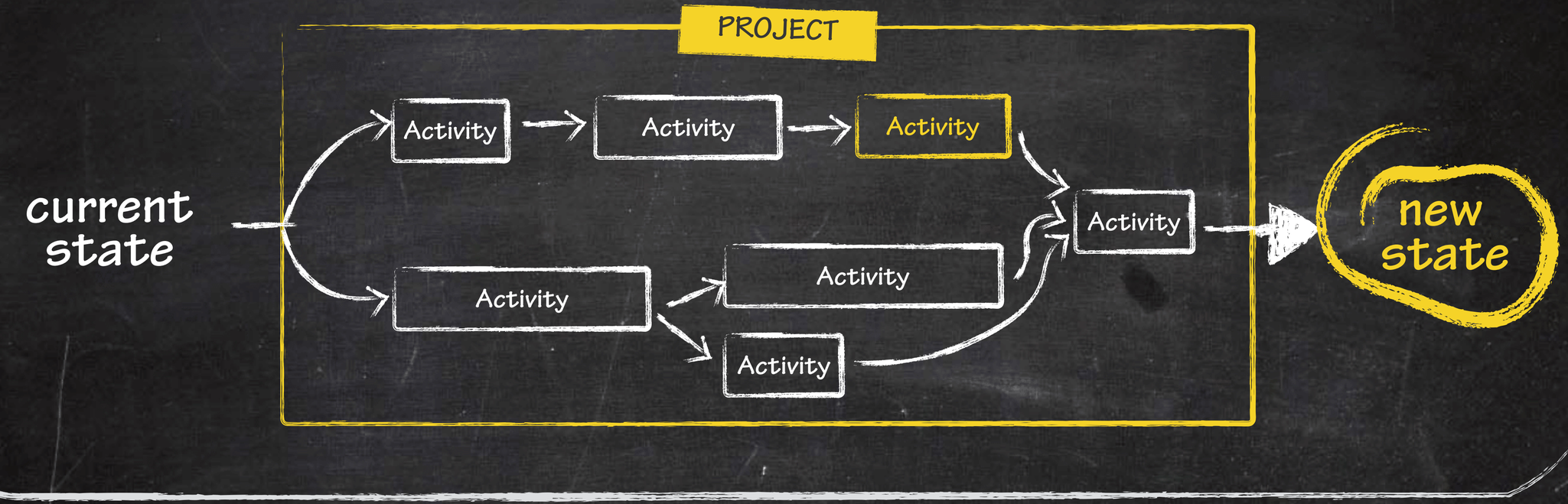
SOFTWARE DEVELOPMENT

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

- well defined **task** to reach a certain **goal**
- Consists of **activities** (of finite duration)



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 whats its Management?

Examples

research

state of the art (literature)

research question is stated

BTW: This *does not* change when carefully formulated!

PROJECT

research question is answered

paper

development

current release

new functionality is needed

BTW: This *will* change since customer's needs always change

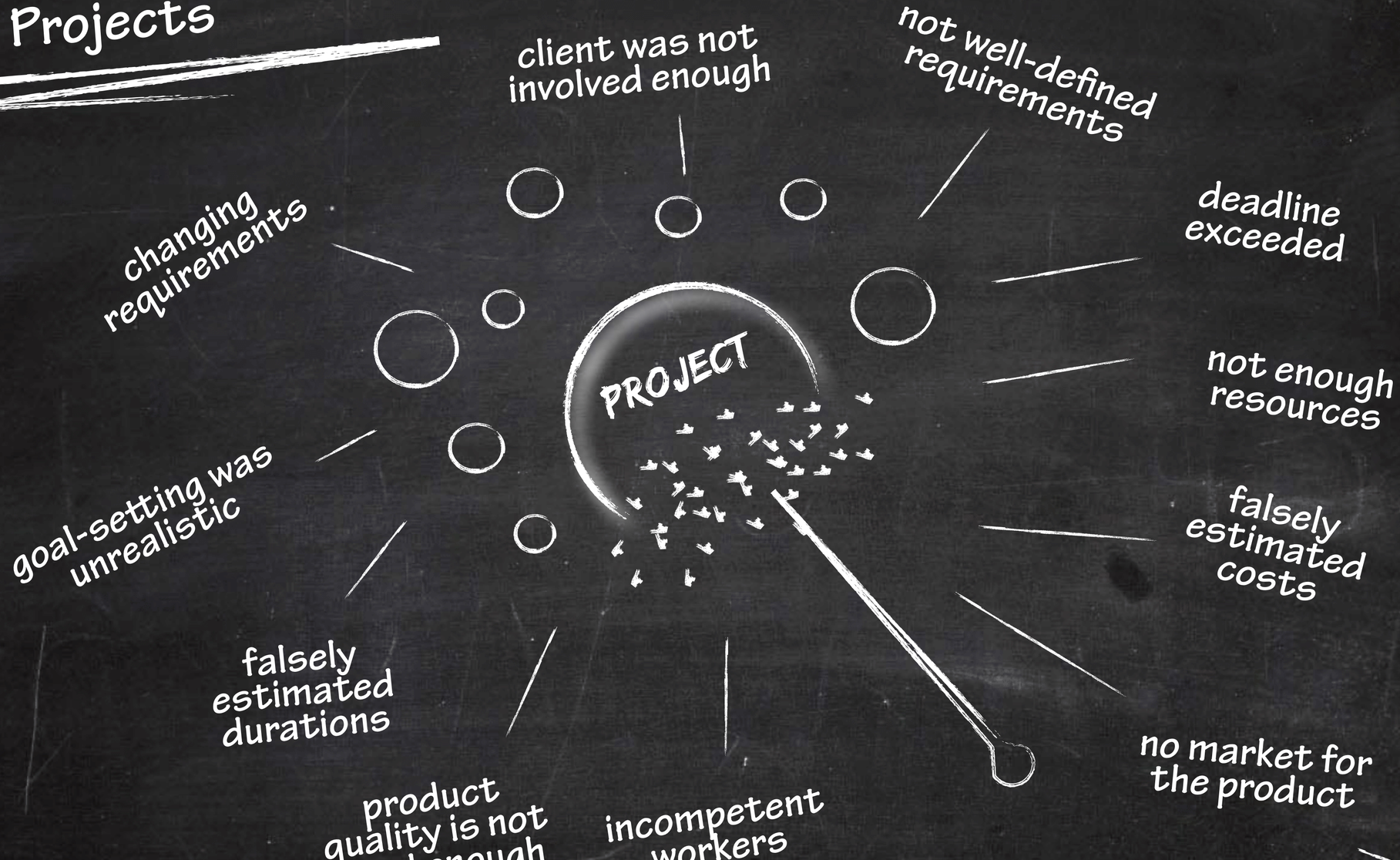
PROJECT

new functionality is available

next release

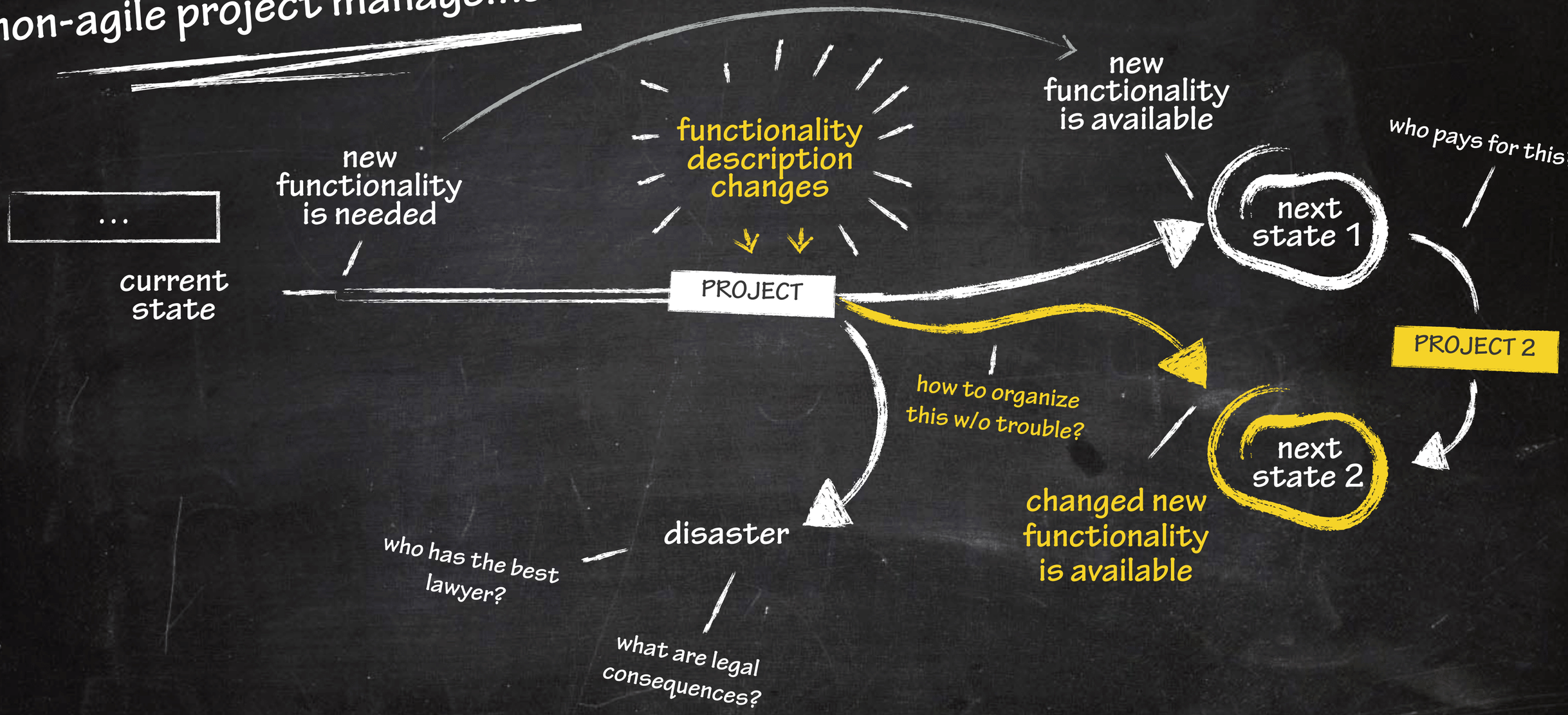
...

Reasons for Failed Projects

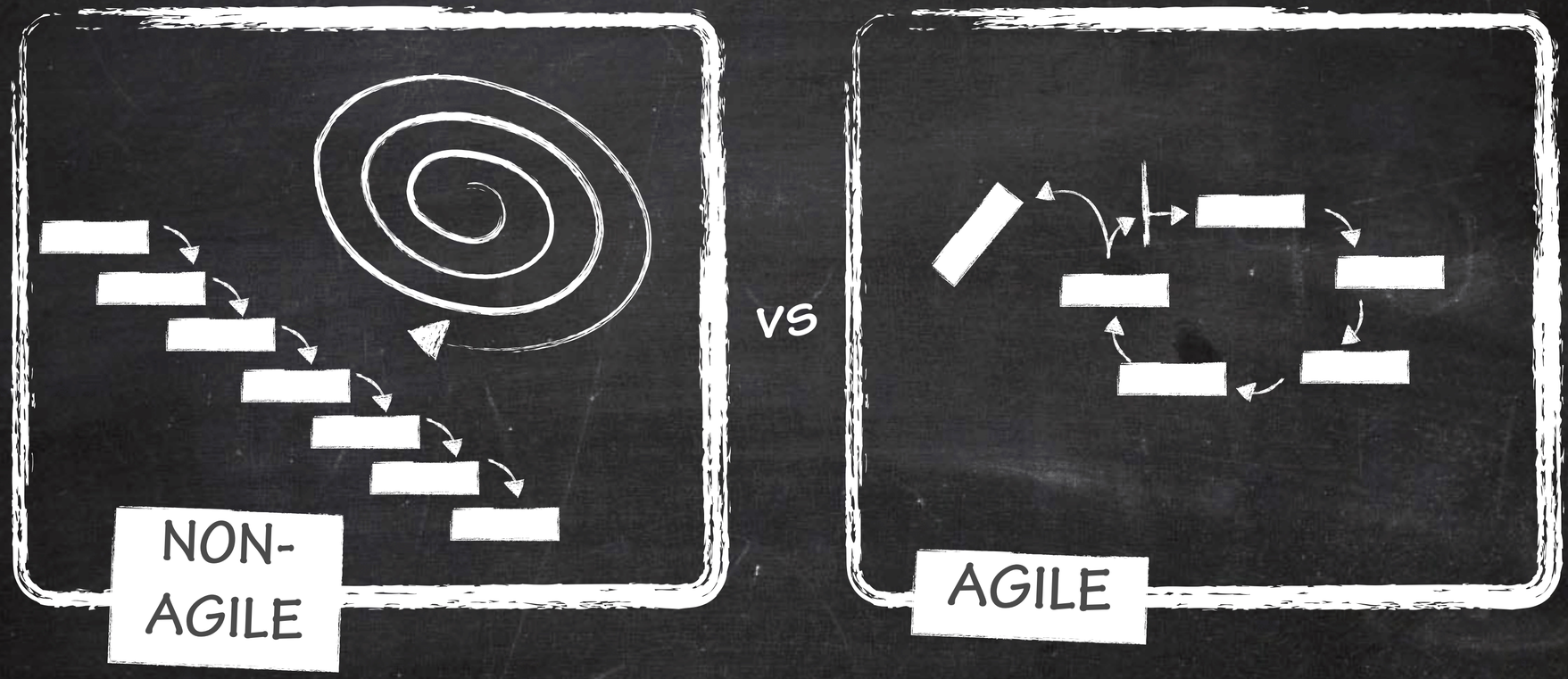


try to minimize these risks

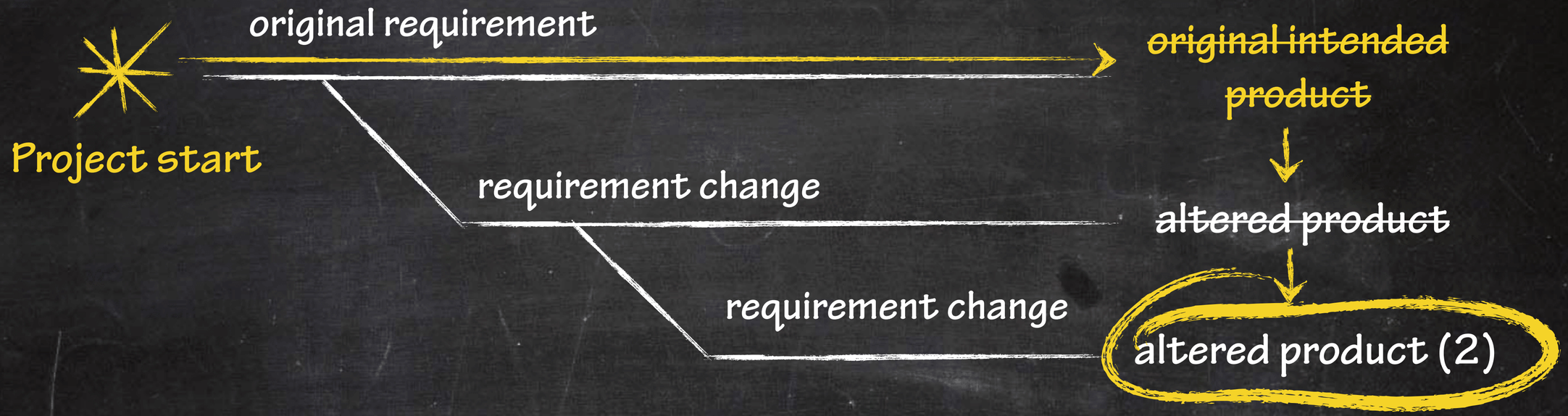
Changes in requirements for non-agile project management



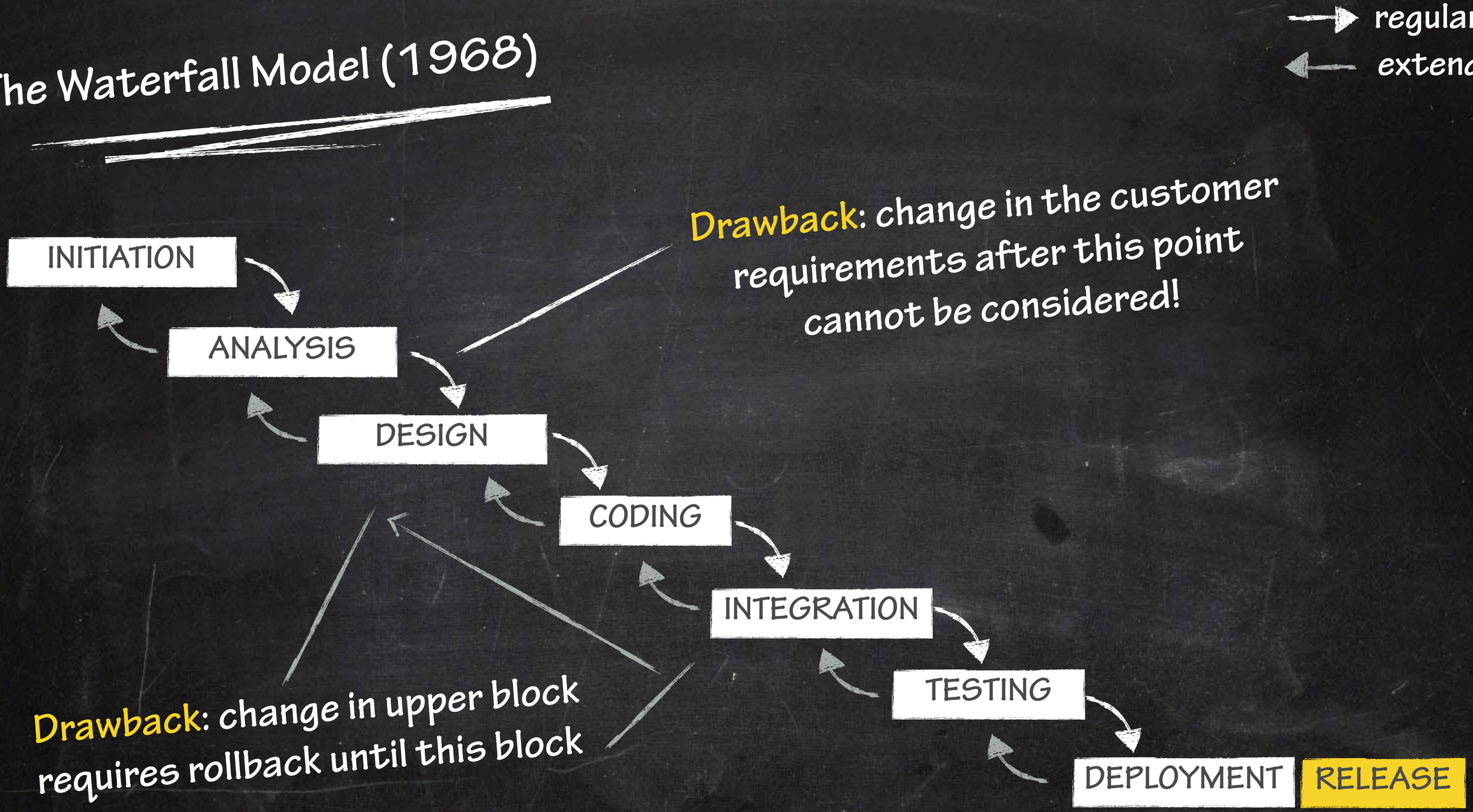
Development Methodologies



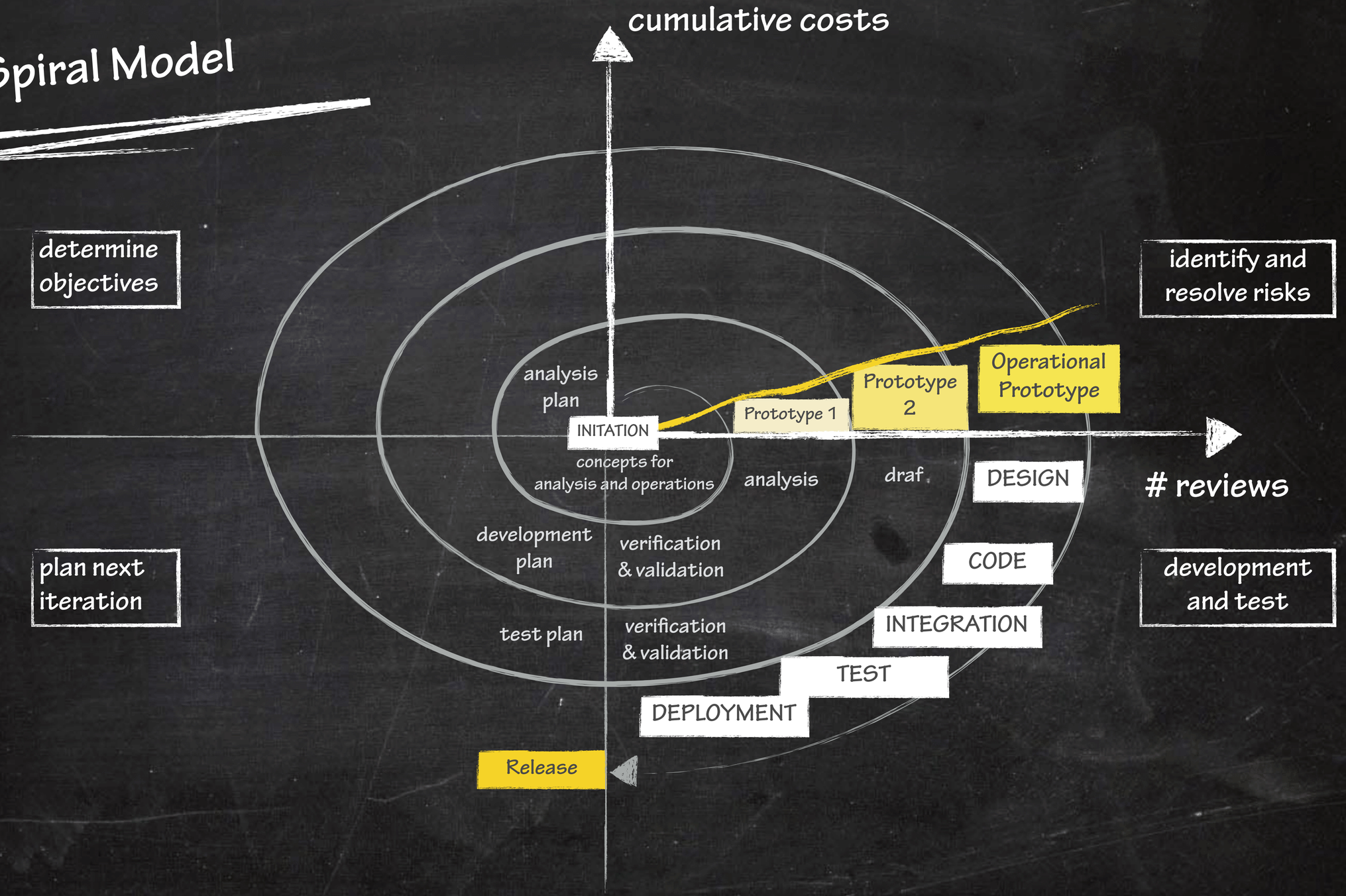
Non-Agile Methodologies



The Waterfall Model (1968)



The Spiral Model



Manifesto for Agile Software Development

Individuals and interactions
over processes and tools

Working software
over comprehensive documentation

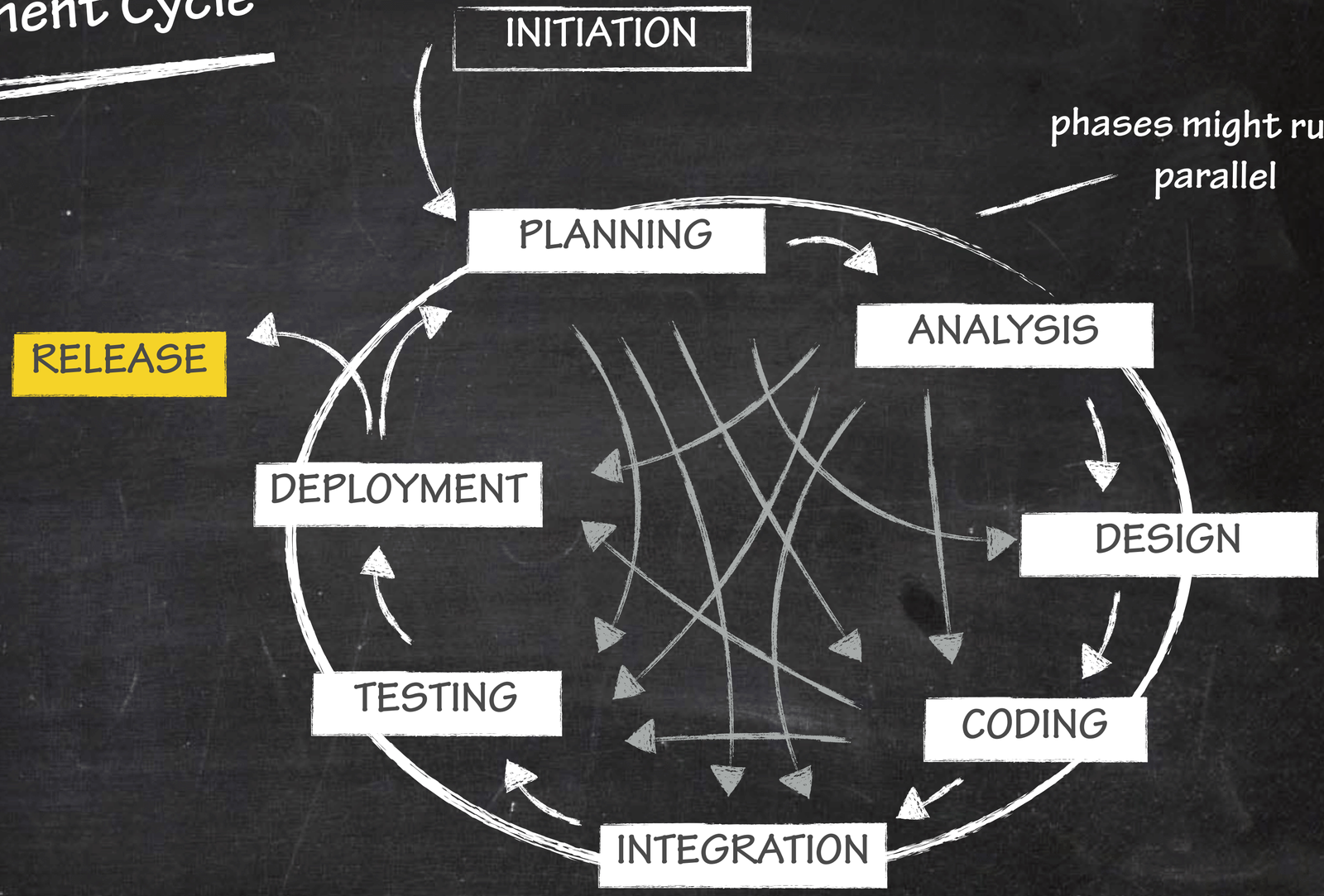
Customer collaboration
over contract negotiation

Responding to change
over following a plan

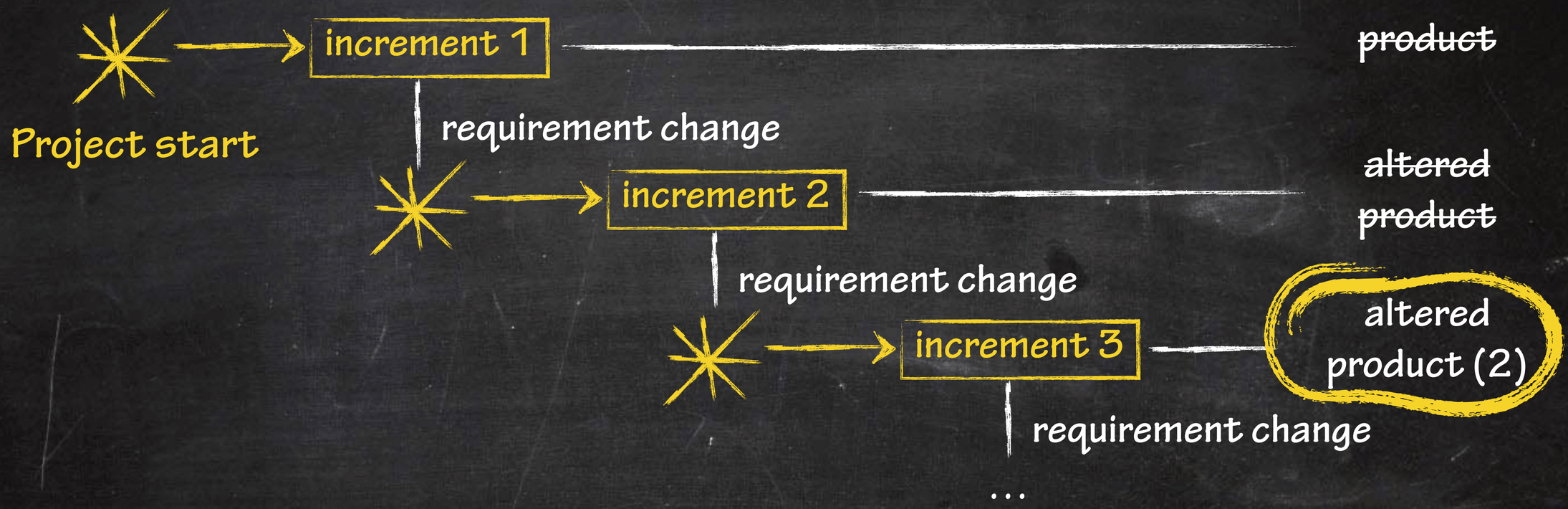
Agile software development is an idea not a concrete method.

http://agilemanifesto.org

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

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

Lean Software Development (LSD)

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

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

Dynamic Systems Development Method (DSDM)

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

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

Extreme Programming (XP)

focus: improve quality and responsiveness to evolving customer requirements

principles: include feedback, assuming simplicity, embracing change

Scrum

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

most popular

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

Adaptive System Development (ASD)

focus: projects should continuous adapt

cycle of 3 activities: speculate, collaborate, learn

Kanban

focus: visualize things for small, continuous changes

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

Feature-Driven Development (FDD)

focus: iterative and incremental based on industry best practice

5 basics: develop overall model, build feature list, plan by feature, design by feature, and build by feature

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