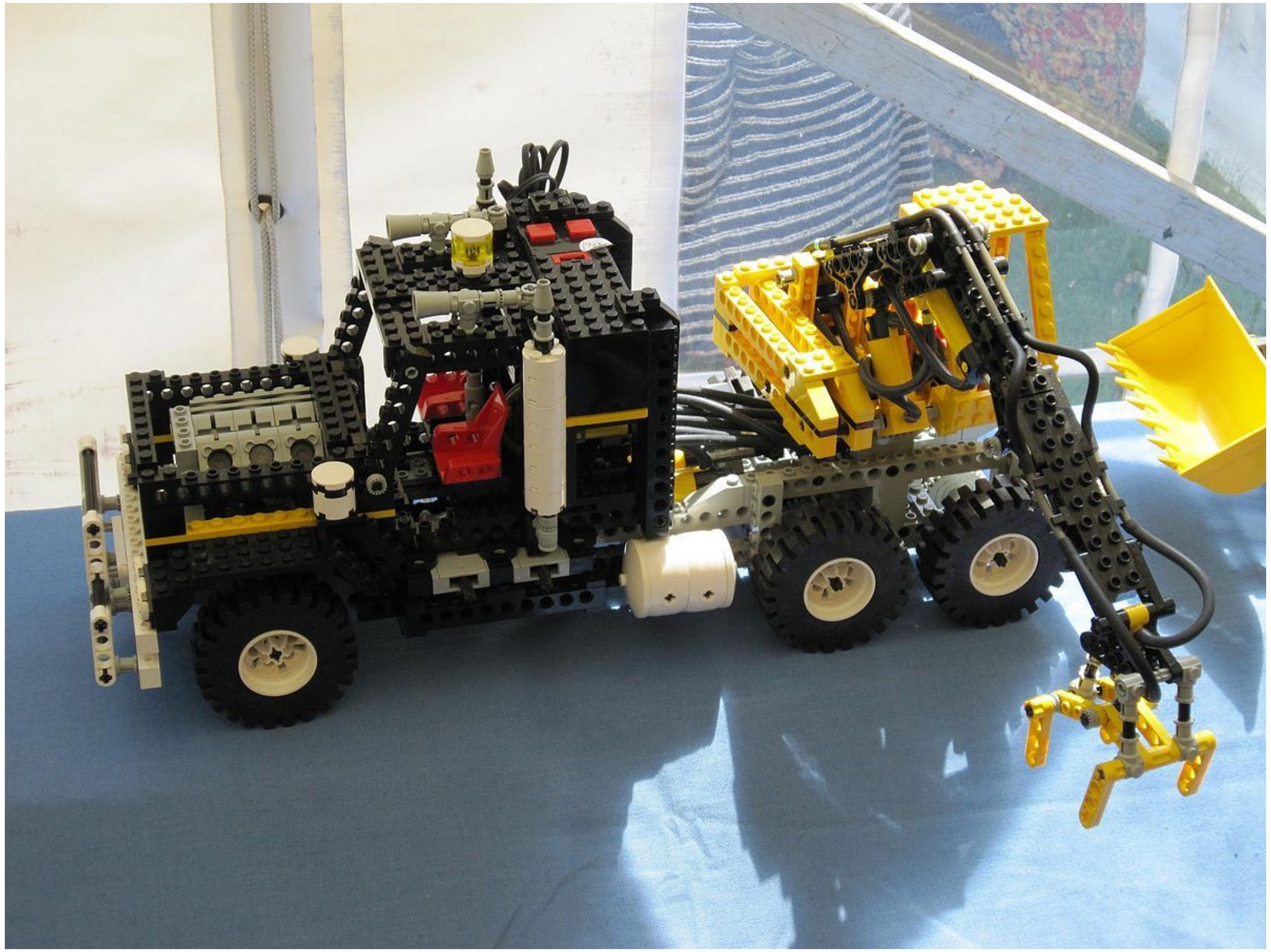
CS 301 Process Models

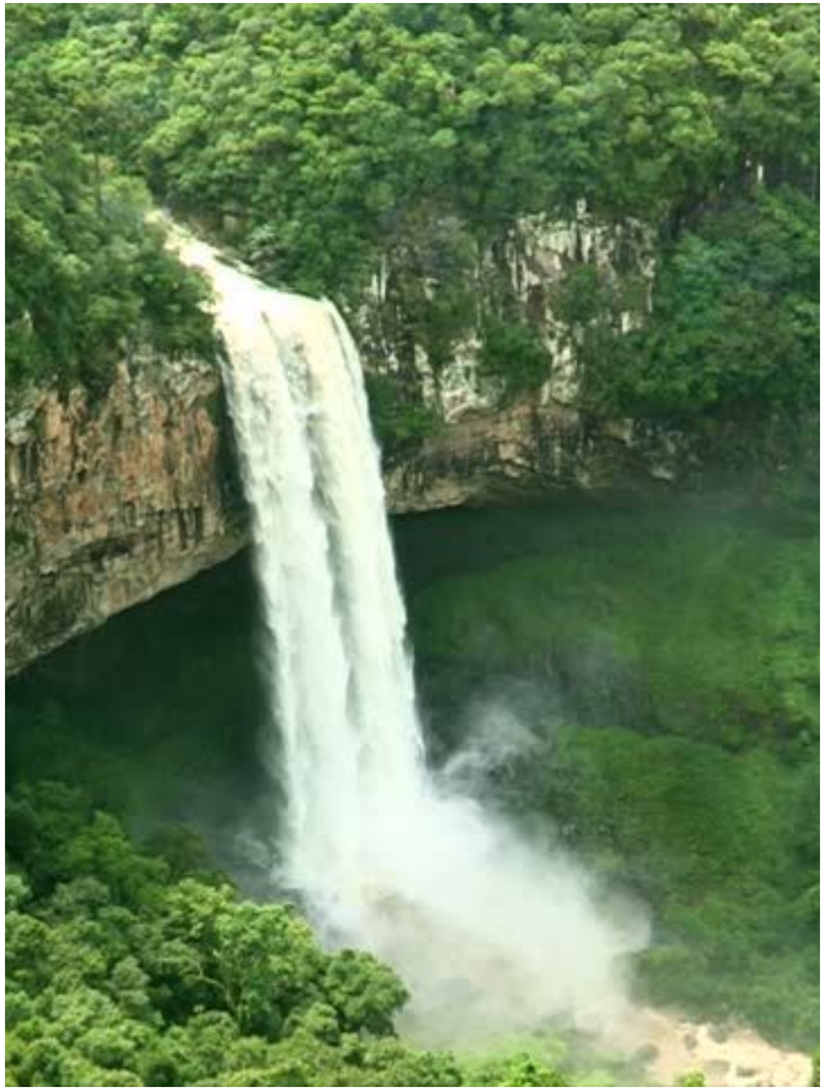
Professor Eswaran

21 Aug 2024

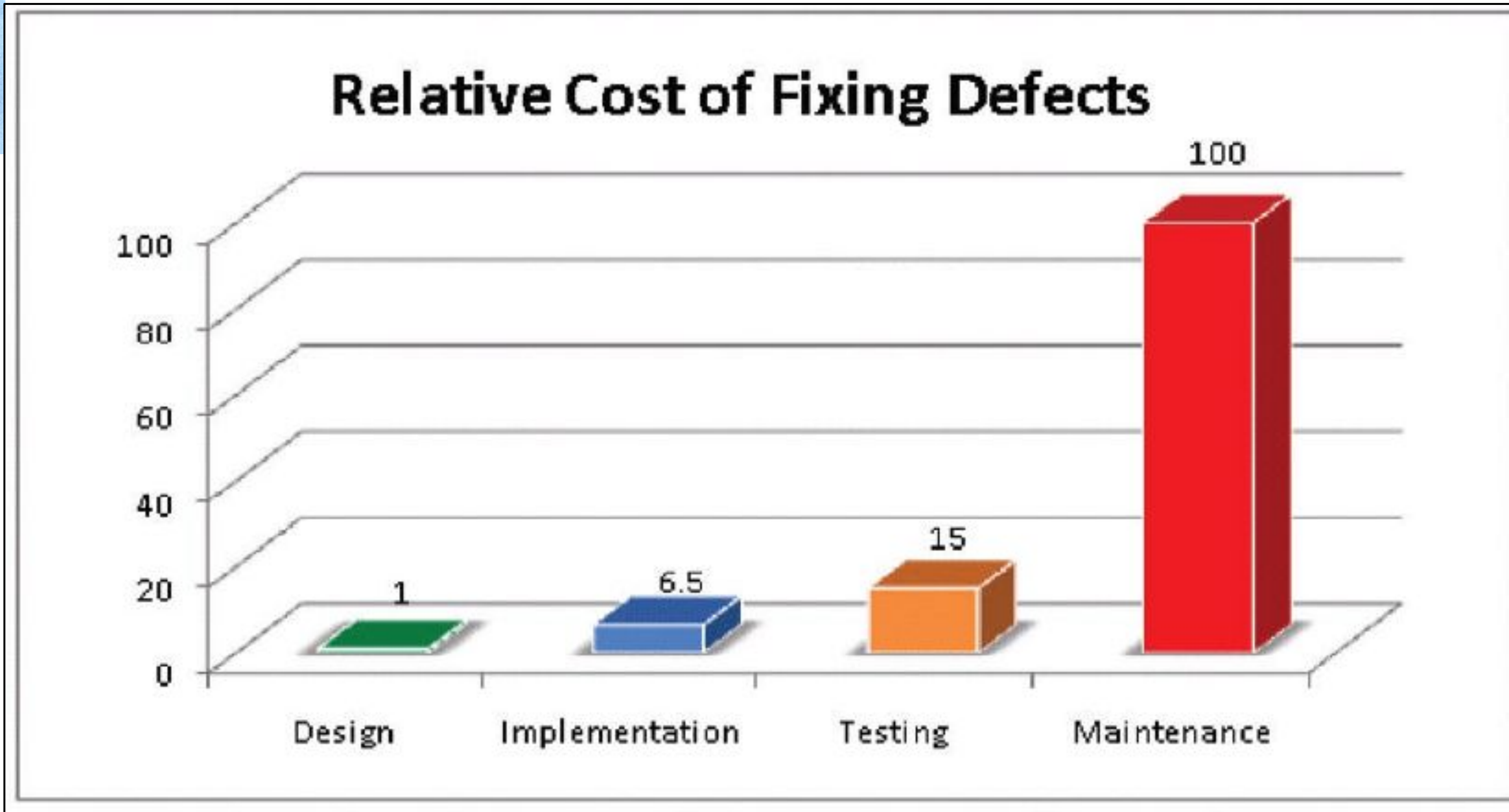
Software Development Methodologies

https://ak3.picdn.net/shutterstock/videos/11611373/thumb/1.jpg





Prototyping



Prototyping

Increased confidence in requirements gathering

- ❑ Crystallize requirements understanding
- ❑ Accepting tea leaf syndrome
- ❑ Demonstrate proof of understanding

Prototyping

Increased confidence in requirements gathering

- ❑ Crystallize requirements understanding
- ❑ Accepting tea leaf syndrome
- ❑ Demonstrate proof of understanding

Types of Prototyping

- ❑ Evolutionary
- ❑ Extreme
- ❑ Incremental
- ❑ Throwaway

Prototyping

- Requirements are fuzzy
- Chunk requirements
- Complete and evolve as understanding improves

Increased confidence in requirements gathering

- ❑ Crystallize requirements understanding
- ❑ Accepting tea leaf syndrome
- ❑ Demonstrate proof of understanding

Types of Prototyping

- ❑ Evolutionary
- ❑ Extreme
- ❑ Incremental
- ❑ Throwaway

Prototyping

Increased confidence in requirements gathering

- □ Crystallize requirements understanding
- □ Accepting tea leaf syndrome
- □ Demonstrate proof of understanding

Types of Prototyping

- □ Evolutionary
- □ Extreme
- □ Incremental
- □ Throwaway

- □ Chunk requirements
- □ Complete each chunk
- □ Typically for Web Applications

Prototyping

Increased confidence in requirements gathering

❑ Crystallize requirements understanding

❑ Accepting tea leaf syndrome

❑ Demonstrate proof of understanding

Types of Prototyping

❑ Evolutionary

❑ Extreme

❑ Incremental

❑ Throwaway

▯ Many stages of Minimum Viable Product

▯ Quick approach

▯ Final MVP of the product is useful

Prototyping

□ Feasibility check
□ Basic requirements agreement
□ Throwaway

Increased confidence in requirements gathering

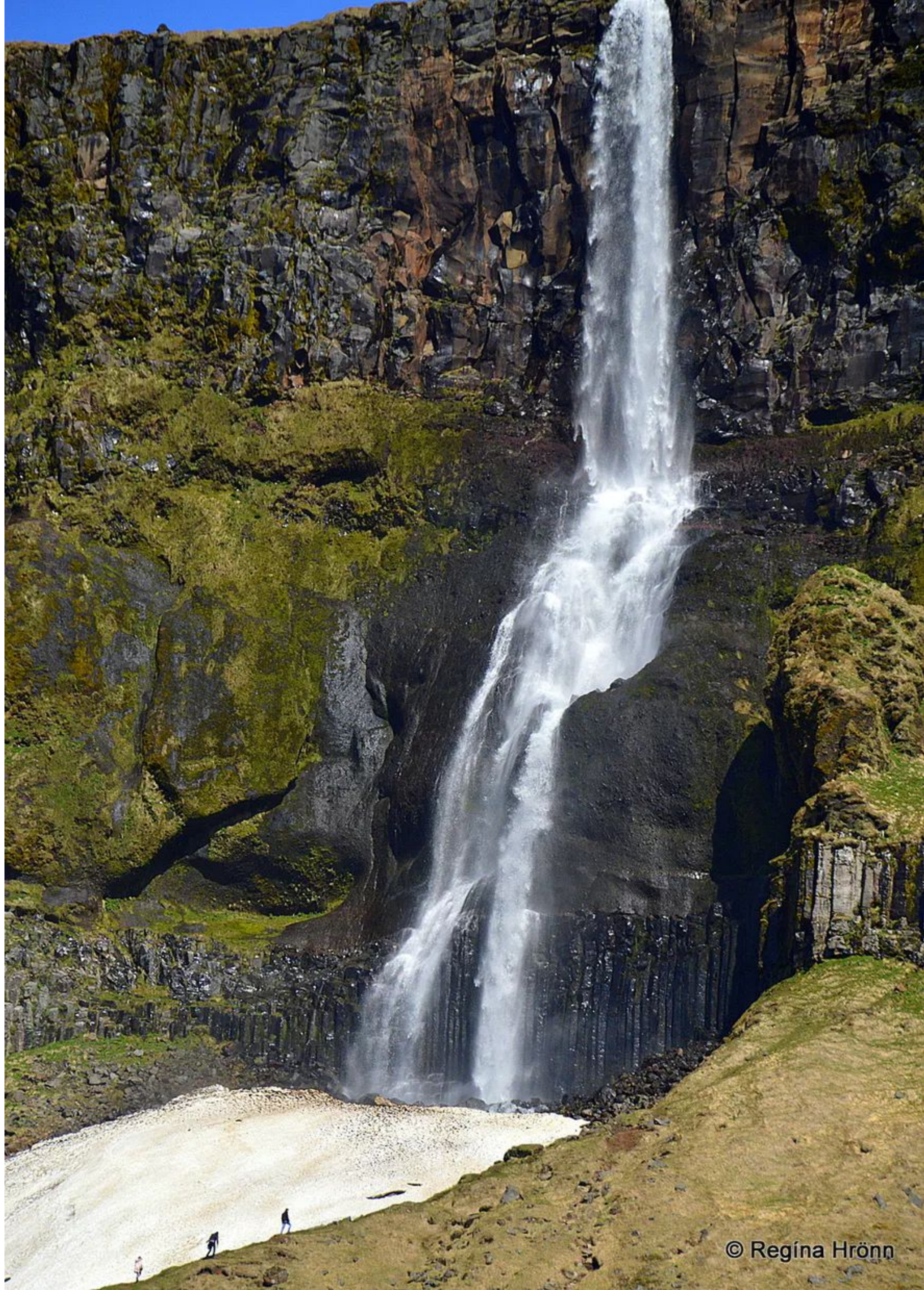
□ Crystallize requirements understanding
□ Accepting tea leaf syndrome
□ Demonstrate proof of understanding

Types of Prototyping

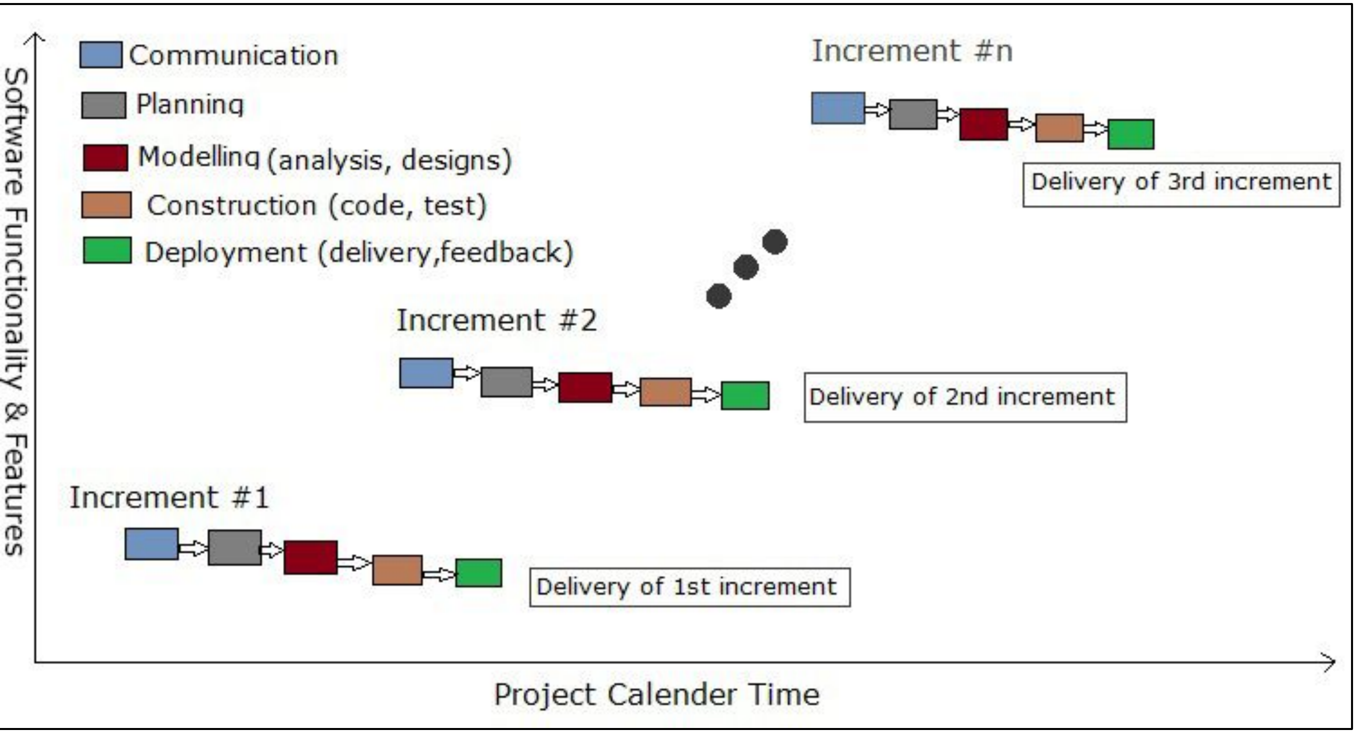
□ Evolutionary
□ Extreme
□ Incremental
□ Throwaway

Incremental Enhancement

https://www.oldtimecandy.com/products/hershey-giant-bar

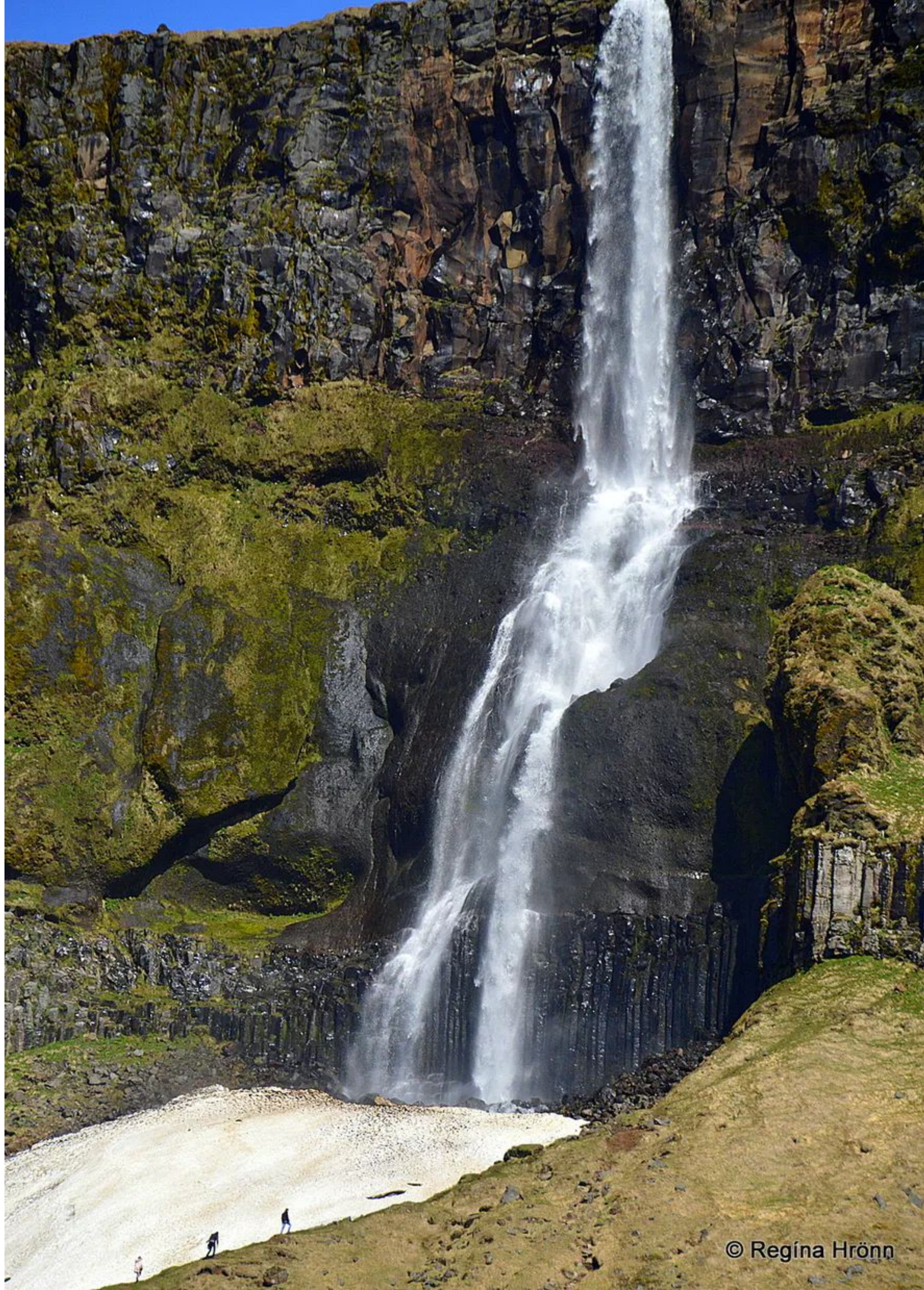


Incremental Enhancement


Communication
Planning
Modelling (analysis, designs)
Construction (code, test)
Deployment (delivery,feedback)
Increment #n
Delivery of 3rd increment
Increment #2
Delivery of 2nd increment
Increment #1
Delivery of 1st increment
Software Functionality & Features
Project Calender Time

https://en.wikipedia.org/wiki/Incremental_build_model#/media/File:Incremental_Model.jpg

❑ Chunk the requirements in manageable parts
❑ Execute the waterfall model
❑ Insert parallelism where possible



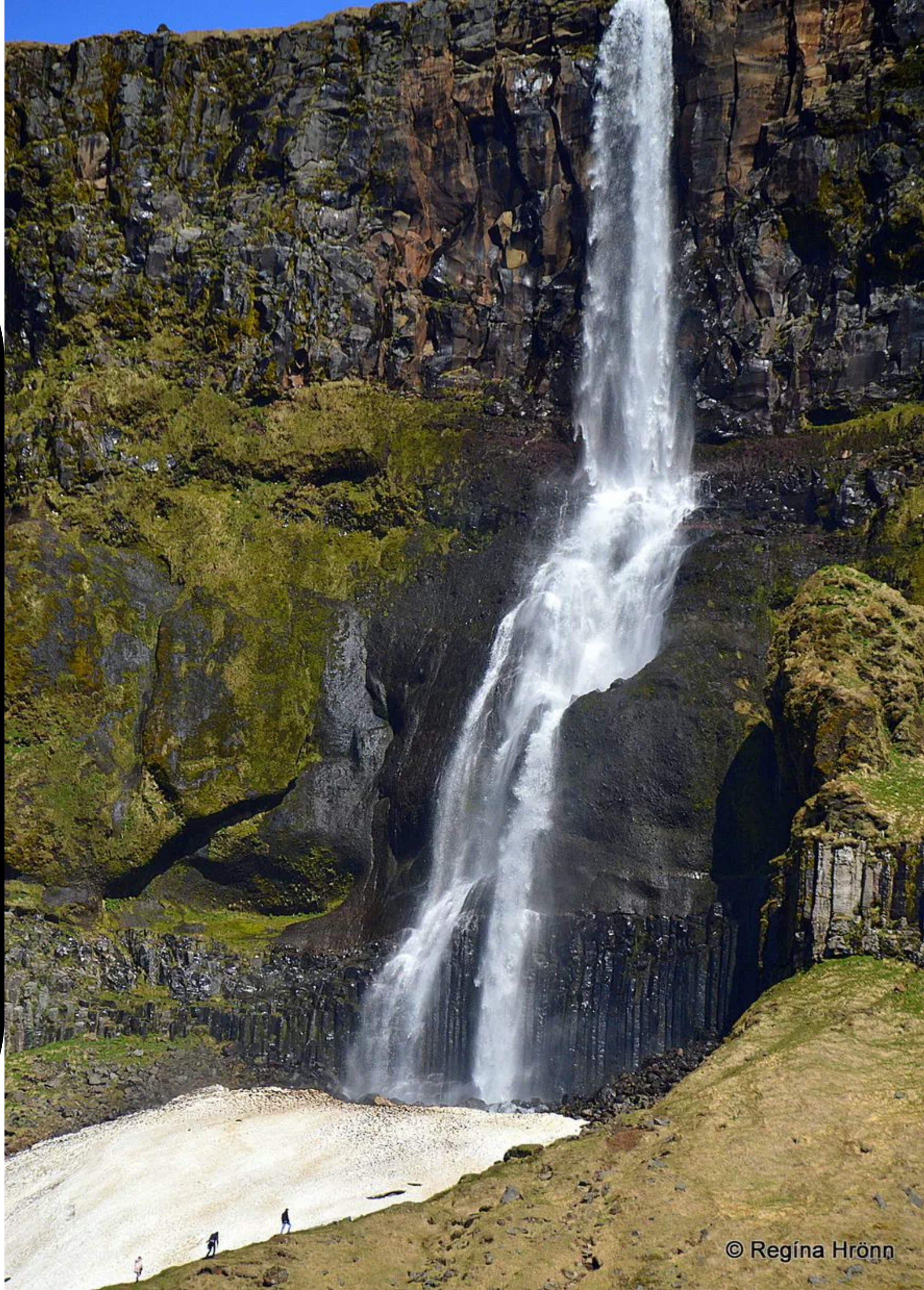
Incremental Enhancement

Advantages

- ❑ Generates working software quickly and early
- ❑ This model is more flexible
- ❑ It is easier to test and debug
- ❑ Feedback cycle is effective
- ❑ Lowers initial cost
- ❑ Risk is Left Shifted
- ❑ Management of people resources easier

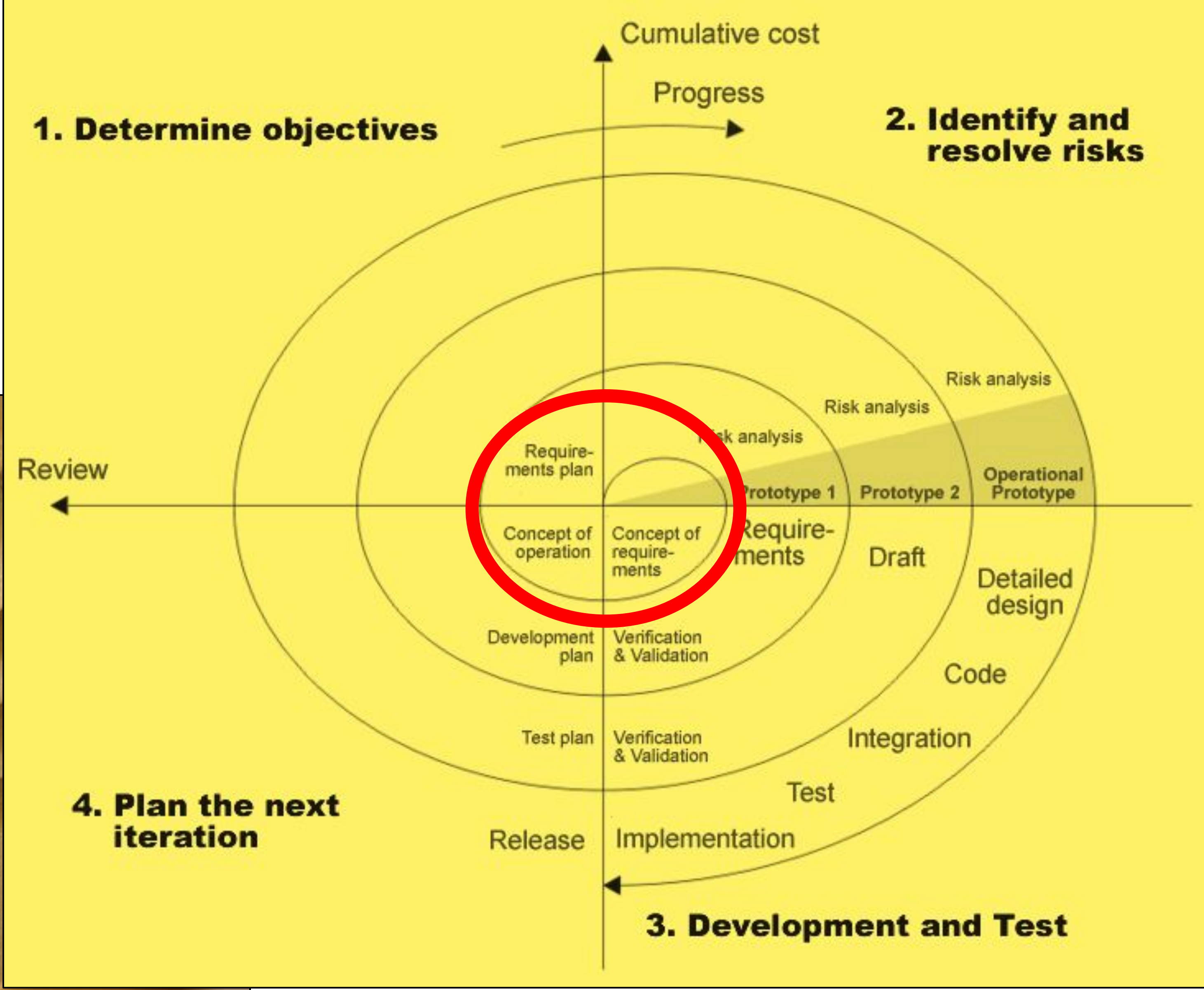
Disadvantages

- ❑ Needs good governance
- ❑ Clarity of Requirements
- ❑ Total cost is higher than waterfall

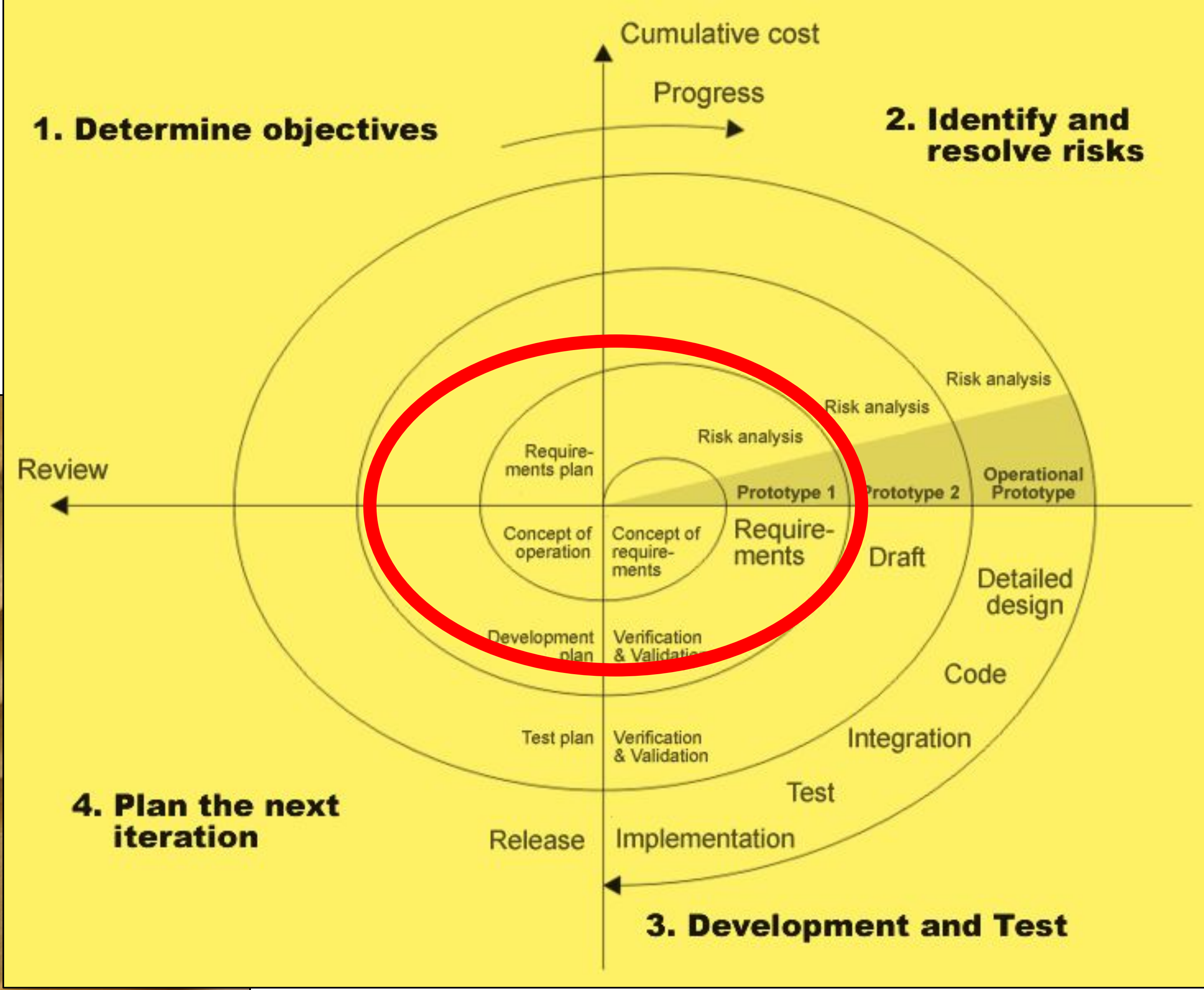


© Regína Hrönn

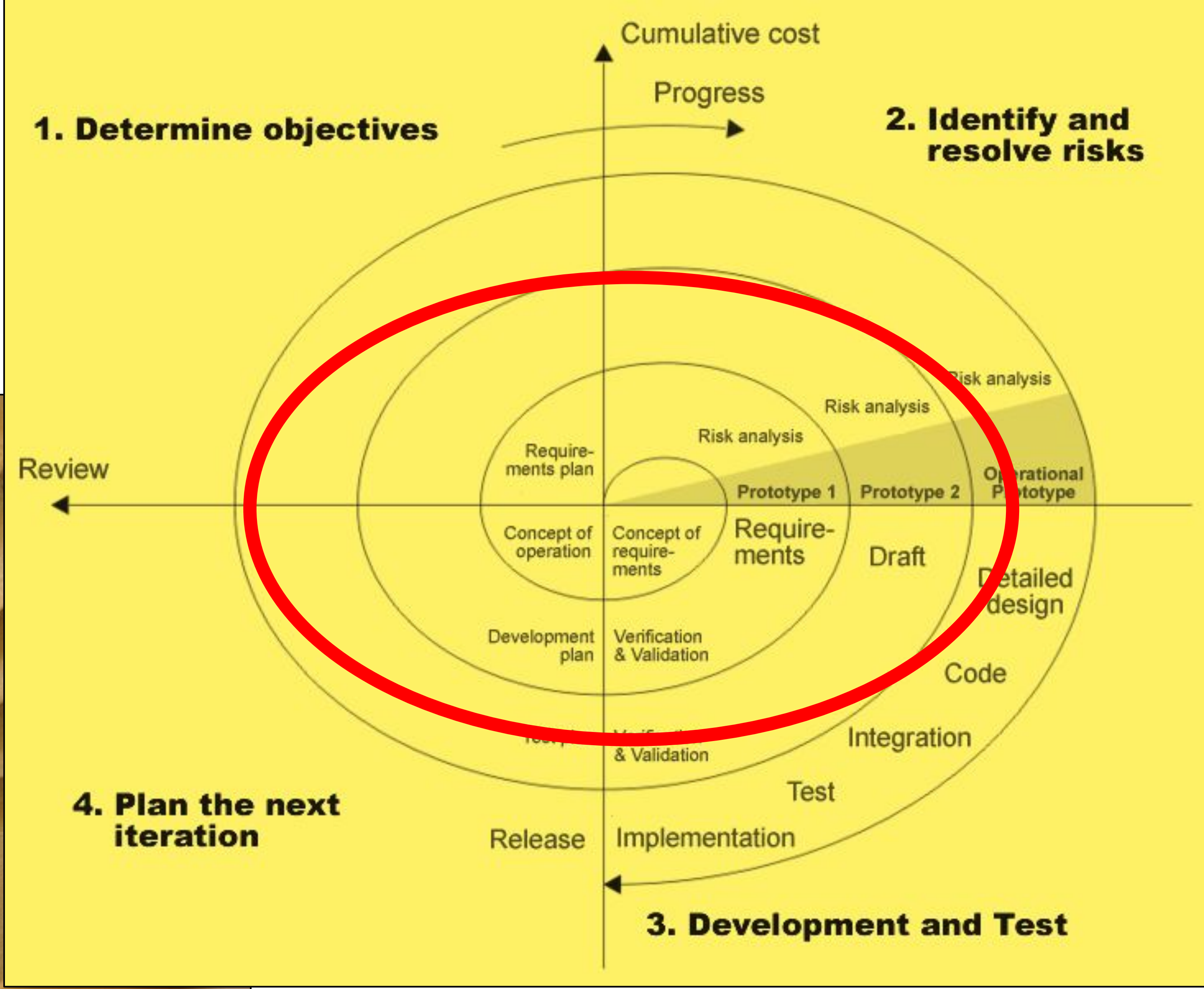
Spiral Model



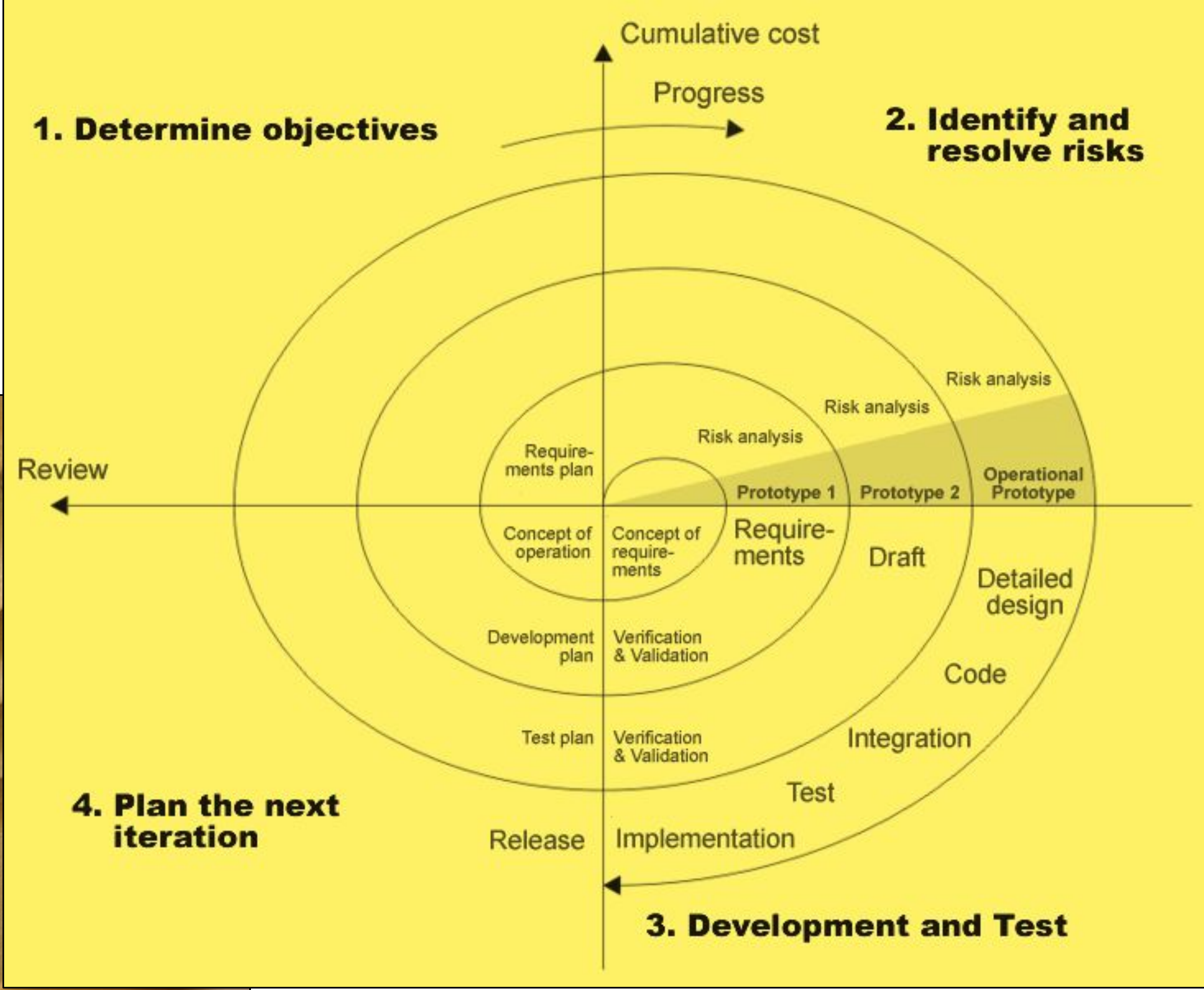
Spiral Model



Spiral Model



Spiral Model



Spiral Model

Advantages

- ❑ Risk Orientation
- ❑ Large and Critical Projects
- ❑ Strong approval and documentation control
- ❑ Feature Additions are deferred

Disadvantages

- ❑ Costs are not easy to control
- ❑ Risk analysis requires highly specific expertise.
- ❑ Success dictated by threat perception
- ❑ Doesn’t work well for smaller projects.

Software Development Methodologies

Properties of Model	Incremental Model	Spiral Model
Planning in early stage		
Returning to an earlier step		
Documentation		
Cost		
Flexibility to change		
User Involvement		
Risk Involvement		
Testing		
Working software availability		
Team size		
Customer control over administrator		

Software Development Methodologies

Properties of Model	Incremental Model	Spiral Model
Planning in early stage	Yes	Yes
Returning to an earlier step	Yes	Yes
Documentation	Yes but not much	Yes
Cost	Low	Expensive
Flexibility to change	Easy	Easy
User Involvement	Intermediate	High
Risk Involvement	Low	Medium to high risk
Testing	After every iteration	At the end of the engineering phase
Working software availability	At the end of every iteration	At the end of every iteration
Team size	Not Large Team	Large Team
Customer control over administrator	Yes	Yes

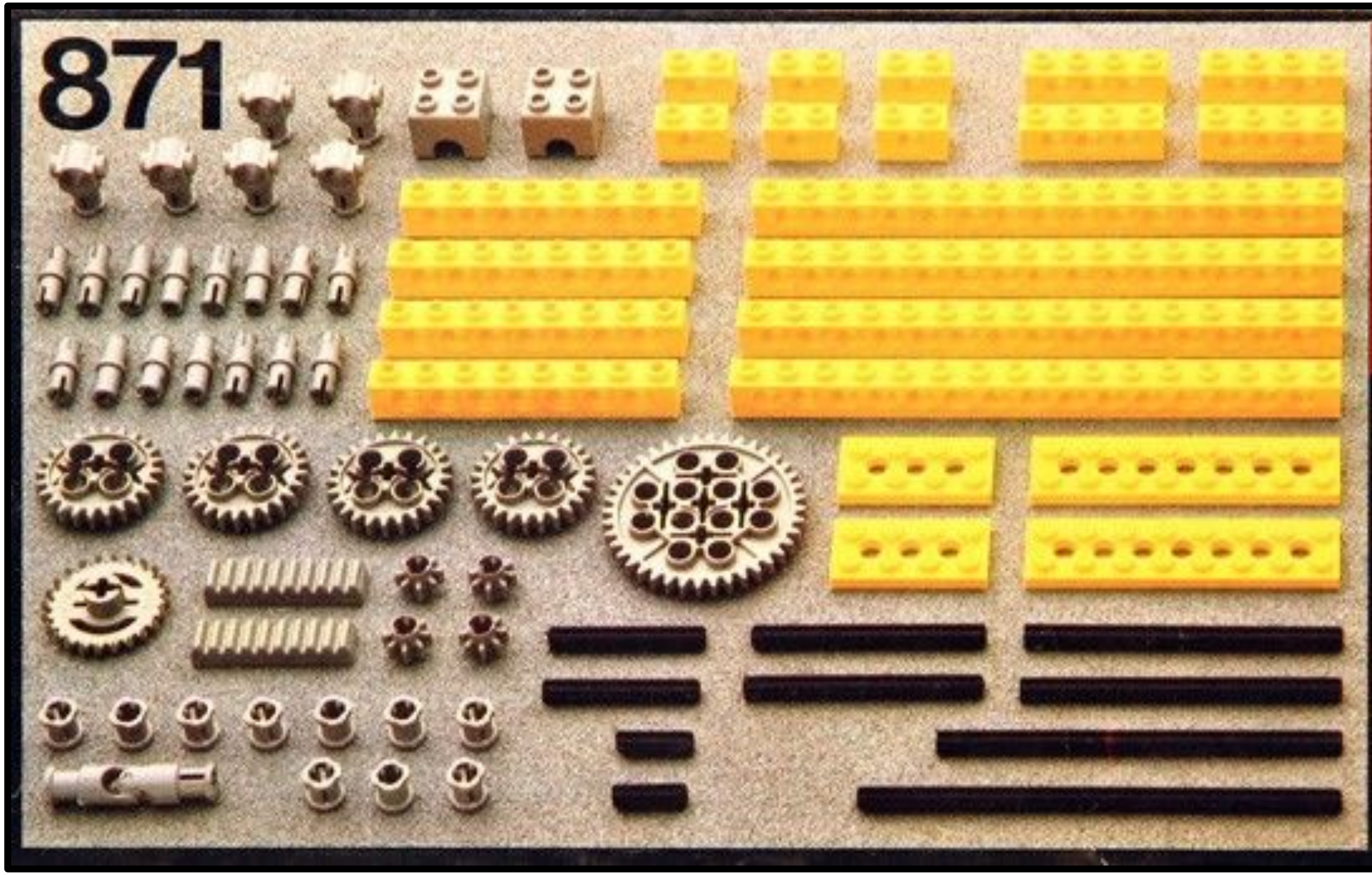
Software Development Methodologies

Properties of Model	Waterfall	Prototyping
Planning in early stage		
Returning to an earlier step		
Documentation		
Cost		
Flexibility to change		
User Involvement		
Risk Involvement		
Testing		
Working software availability		
Team size		
Customer control over administrator		

Software Development Methodologies

Properties of Model	Waterfall	Prototyping
Planning in early stage	Yes – at beginning	Plan at every iteration
Returning to an earlier step	Difficult	Quite possible - encouraged
Documentation	Necessary, Sign off	Lesser Documentation
Cost	High	Relatively indeterminate
Flexibility to change	Low	high
User Involvement	Only in part – Req/Test	Continuous
Risk Involvement	Risk – lesser information	Lower
Testing	After completion of development	Continuous
Working software availability	Absolute End	MVP – available in parts
Team size	Large	Lesser
Customer control over administrator	Less	High

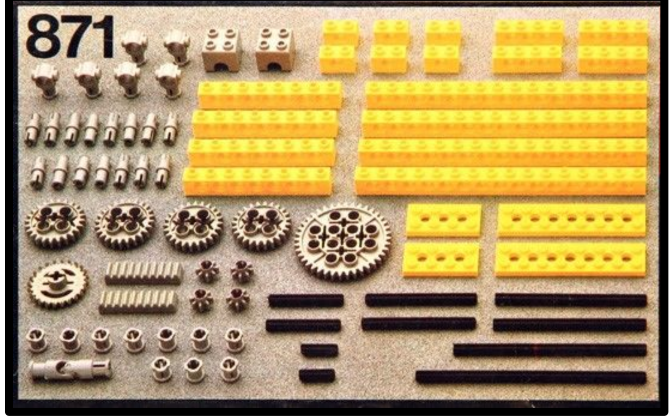
Software Development Methodologies

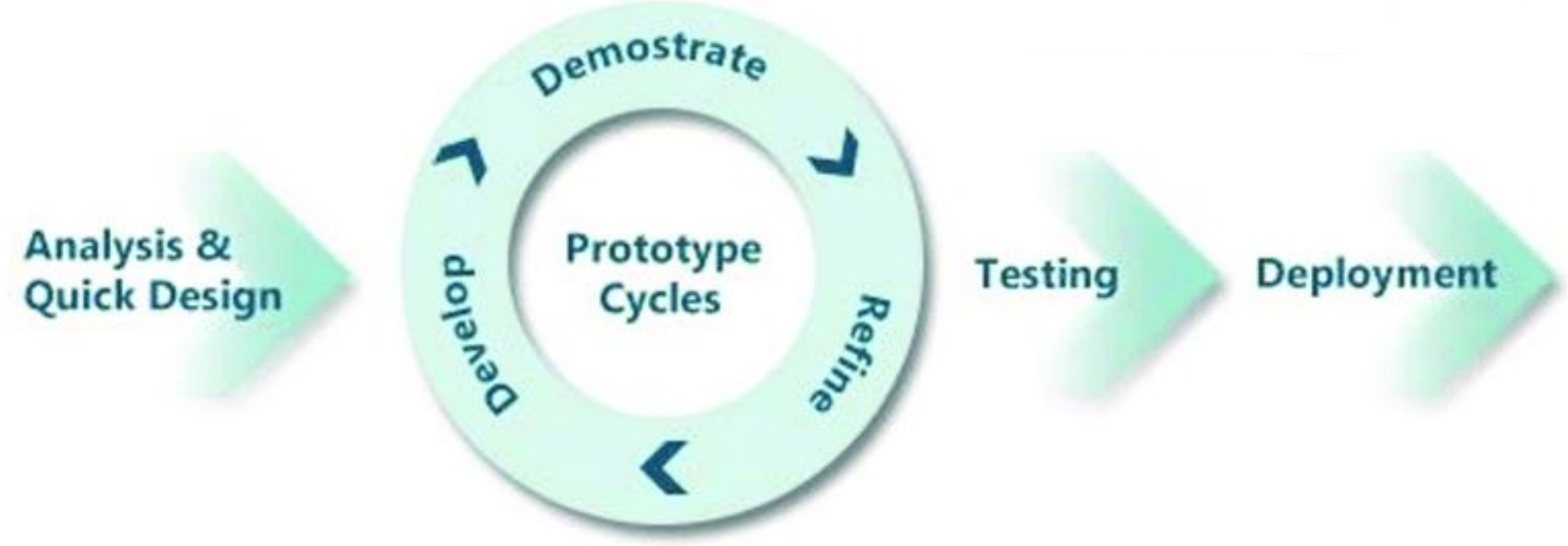


https://brickset.com/sets/961-1/Parts-Pack

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack

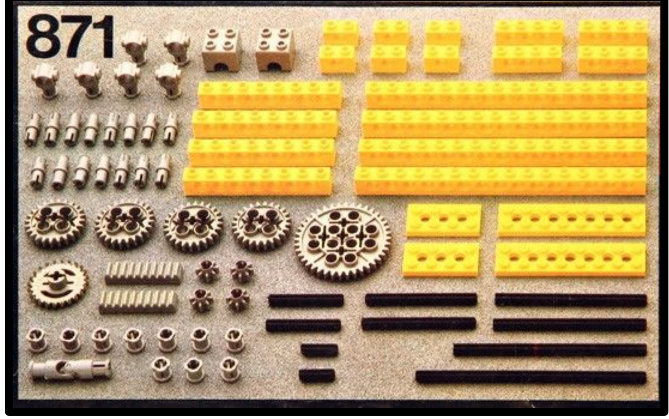


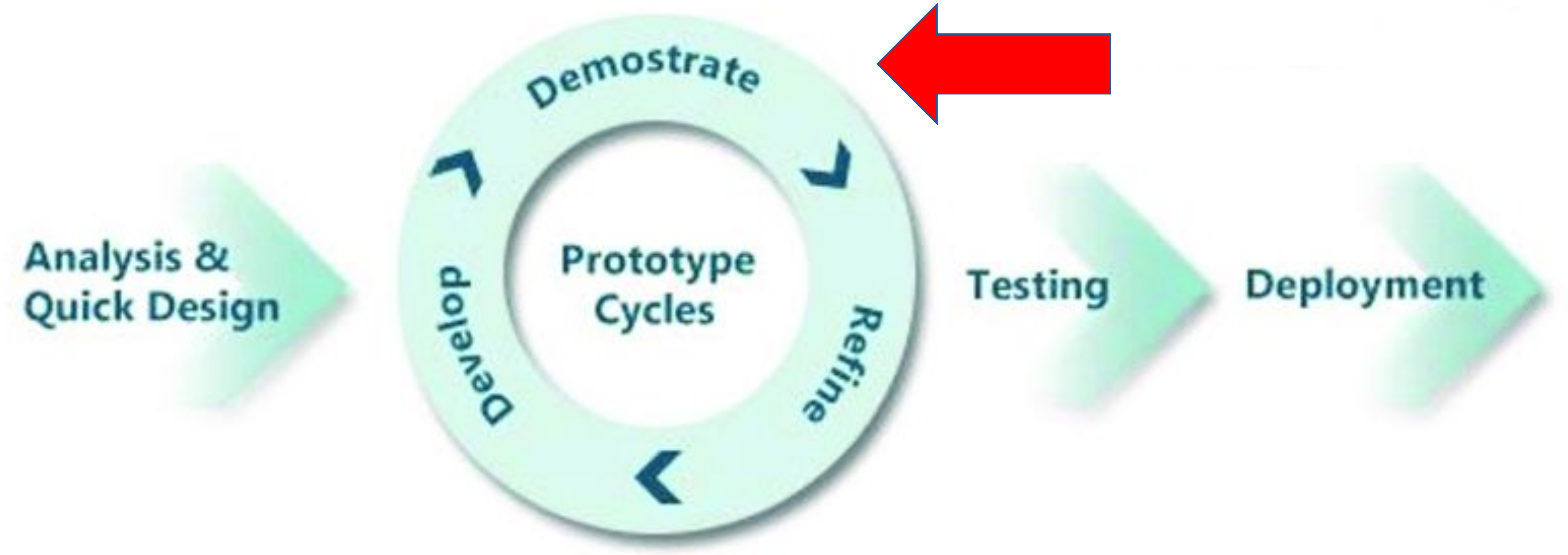


https://www.researchgate.net/figure/Rapid-Application-Development-RAD_fig1_316546533

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack

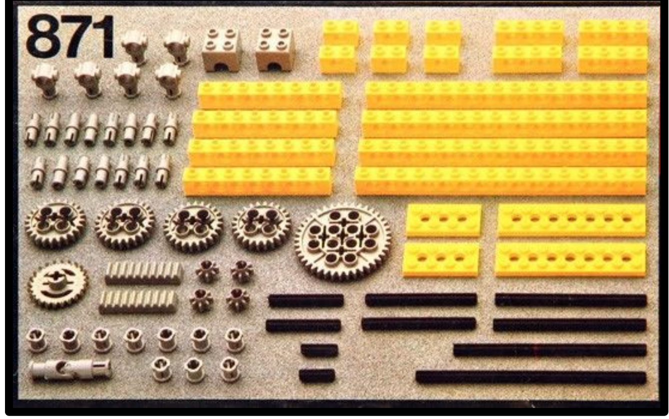


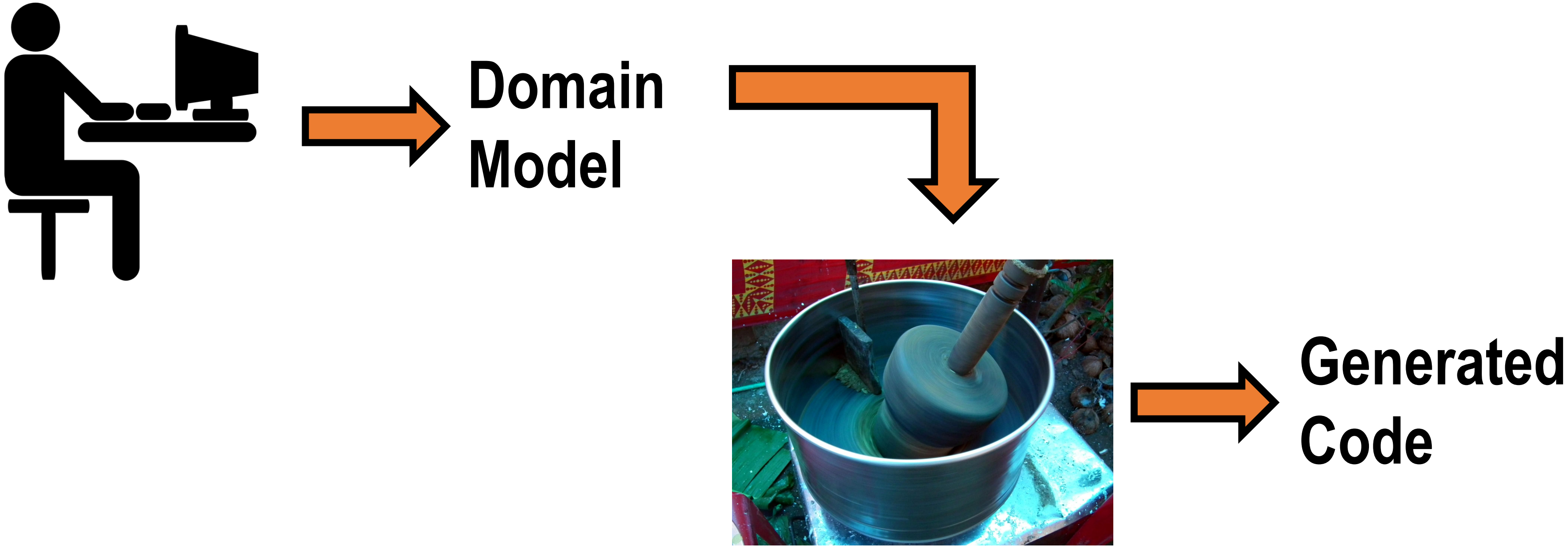


https://www.researchgate.net/figure/Rapid-Application-Development-RAD_fig1_316546533

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack



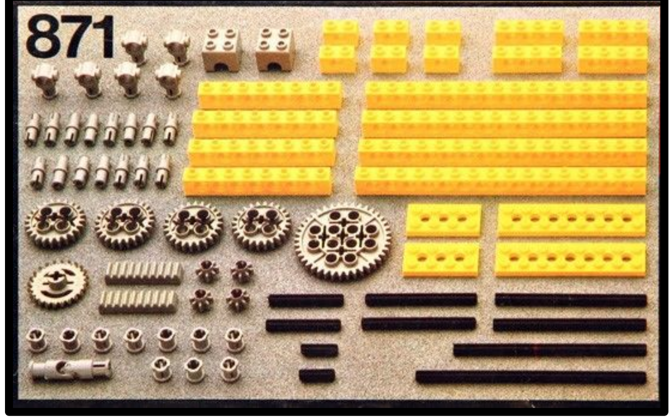


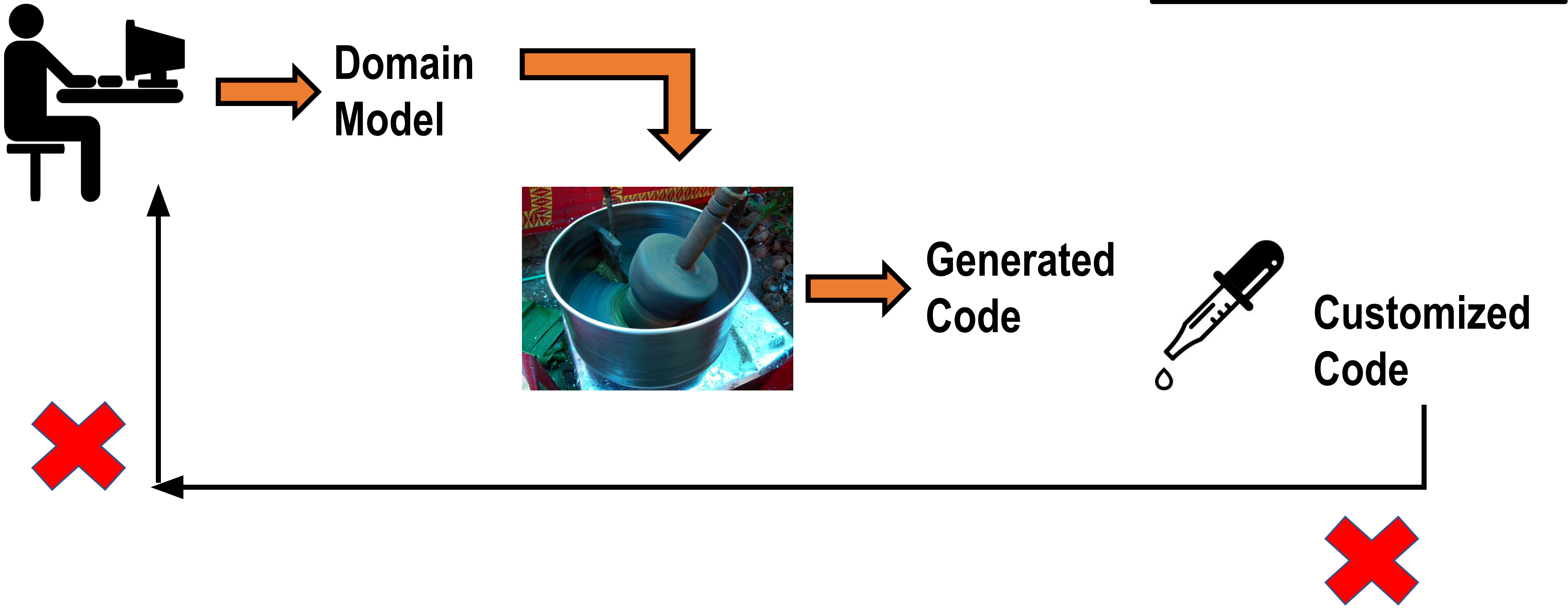
Domain Model

Generated Code

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack





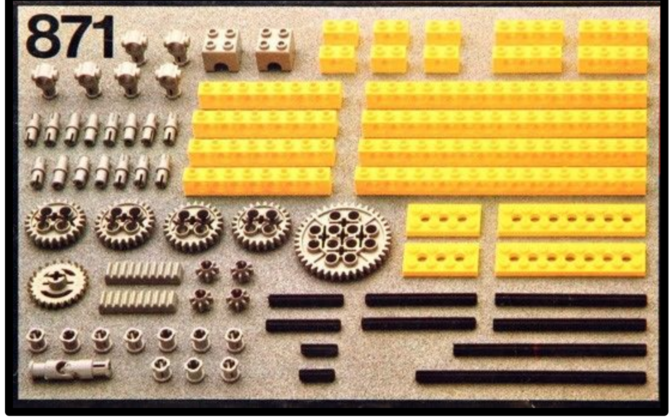
Domain Model

Generated Code

Customized Code

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack



Advantages

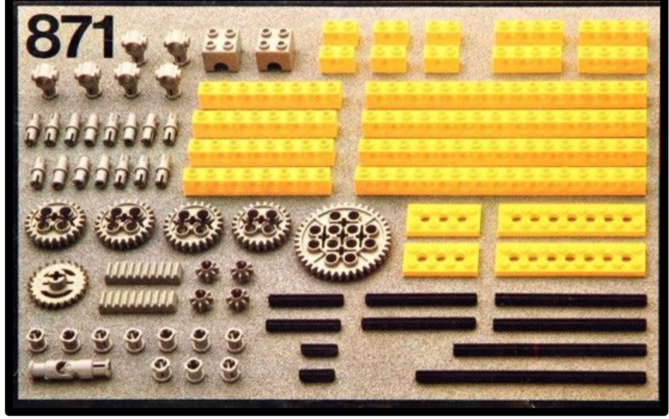
- □ Focus on automation
- □ Feedback from client
- □ Facilitates Go/no go decisions
- □ Risk Control

Disadvantages

- □ Team dependency
- □ Modularization is required in high order
- □ Not recommended for the small budgeted projects
- □ Reverse engineering may be difficult

Rapid Application Development

https://brickset.com/sets/961-1/Parts-Pack



Properties of Model	RAD
Planning in early stage	No
Returning to an earlier step	Yes
Documentation	Low
Cost	Low
Flexibility to change	Easy
User Involvement	Beginning
Risk Involvement	Low
Testing	After Coding
Working software availability	End of Life Cycle
Team size	Small
Customer control over administrator	Yes