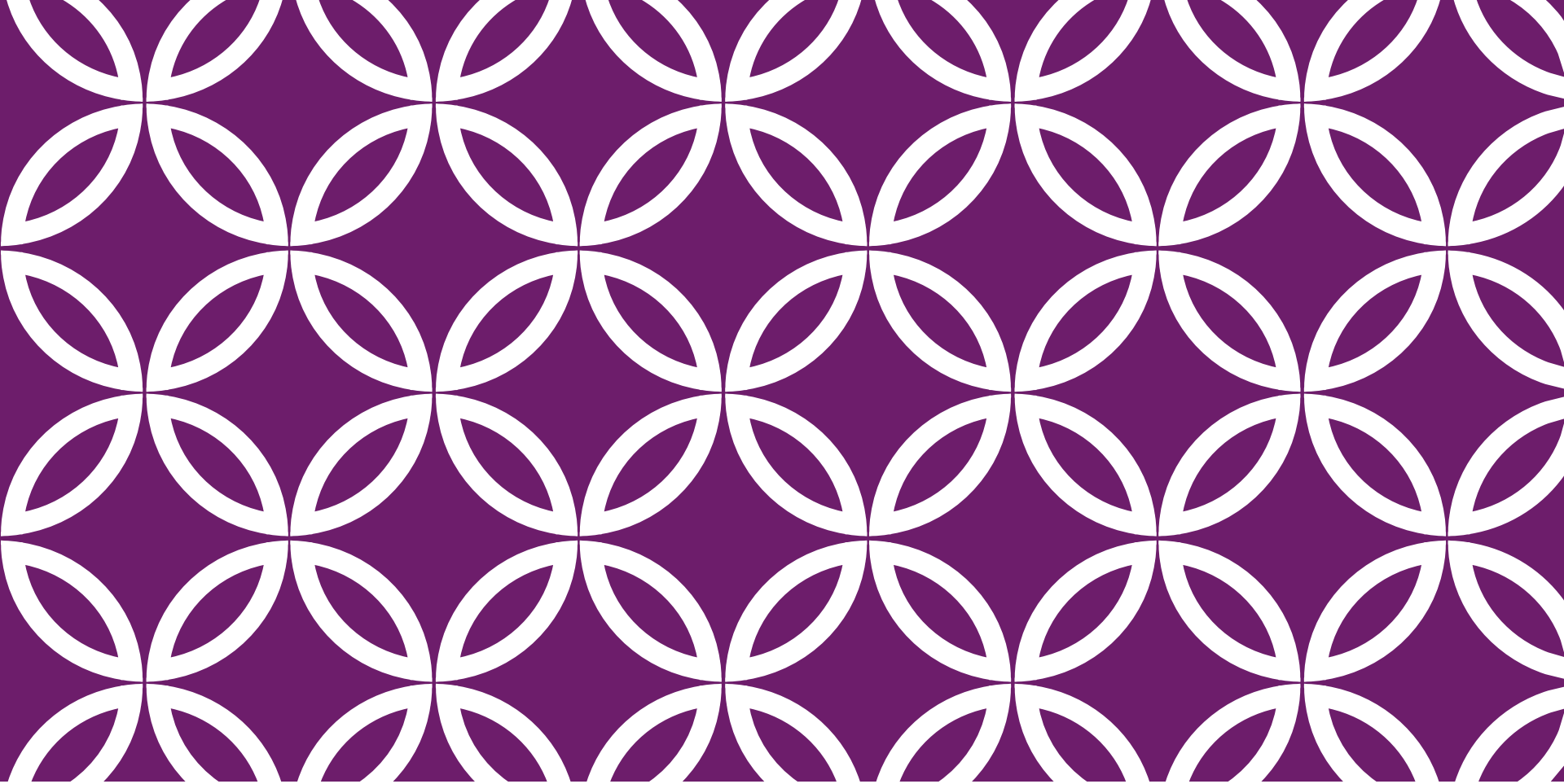
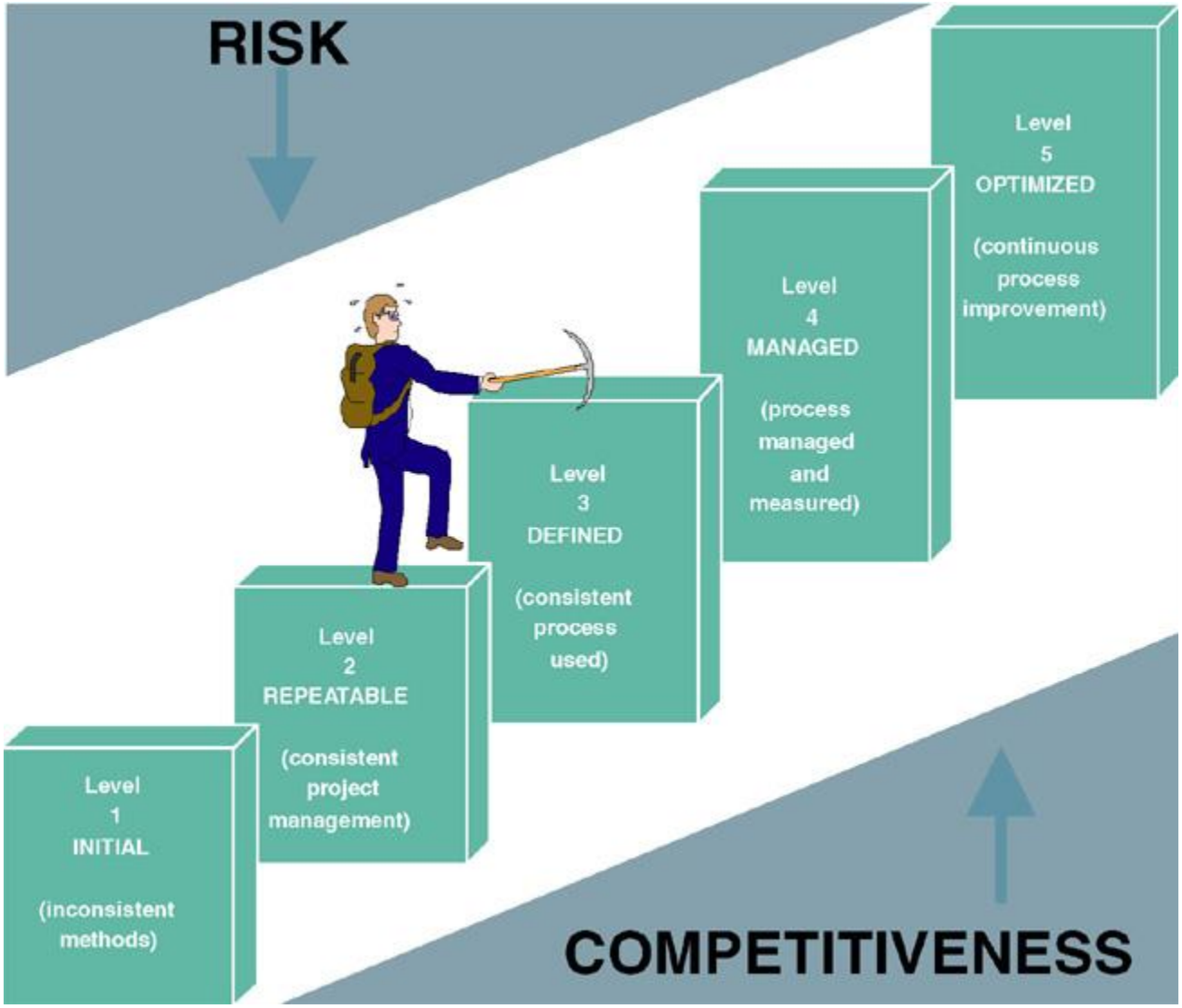


SOFTWARE DEVELOPMENT METHODOLOGIES

Systems Analysis and Design
Sharif University of Technology
Fall 1400-1401

Capability Maturity Model (CMM)



CMM Process Management Model

Capability Maturity Model (CMM) – a standardized framework for assessing the maturity level of an organization's information system development and management processes and products. It consists of five levels of maturity:

- **Level 1—Initial**: System development projects follow no prescribed process.
- **Level 2—Repeatable**: Project management processes and practices established to track project costs, schedules, and functionality.
- **Level 3—Defined**: Standard system development process (methodology) is purchased or developed. All projects use a version of this process.
- **Level 4—Managed**: Measurable goals for quality and productivity are established.
- **Level 5—Optimizing**: The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

Development Methodology

- A Formalized approach to the systems development process;
- A standardized development process that defines (as in CMM Level 3) a set of
  - → activities,
  - → methods,
  - → best practices,
  - → deliverables, and
  - → automated tools
- that system developers and *project managers* are to use to *develop* and continuously *improve* information systems and software.

Development Methodology

Approaches

- Structured
  - Waterfall development
  - Parallel development
  - V-model
- Rapid application development (RAD)
  - Phased
  - Prototyping
  - Throwaway Prototyping
- Agile development
  - Scrum
  - eXtreme Programming (XP)
  - DSDM
  - DAD
  - …

Development Methodology

Approaches

- Structured
  - Waterfall development
  - Parallel development
  - V-model
- Rapid application development (RAD)
  - Phased
  - Prototyping
  - Throwaway Prototyping
- Agile development
  - Scrum
  - eXtreme Programming (XP)
  - DSDM
  - DAD
  - …

Development Methodology

Approaches

- Structured
  - Waterfall development
  - Parallel development
  - V-model
- Rapid application development (RAD)
  - Phased
  - Prototyping
  - Throwaway Prototyping
- Agile development
  - Scrum
  - eXtreme Programming (XP)
  - DSDM
  - DAD
  - …

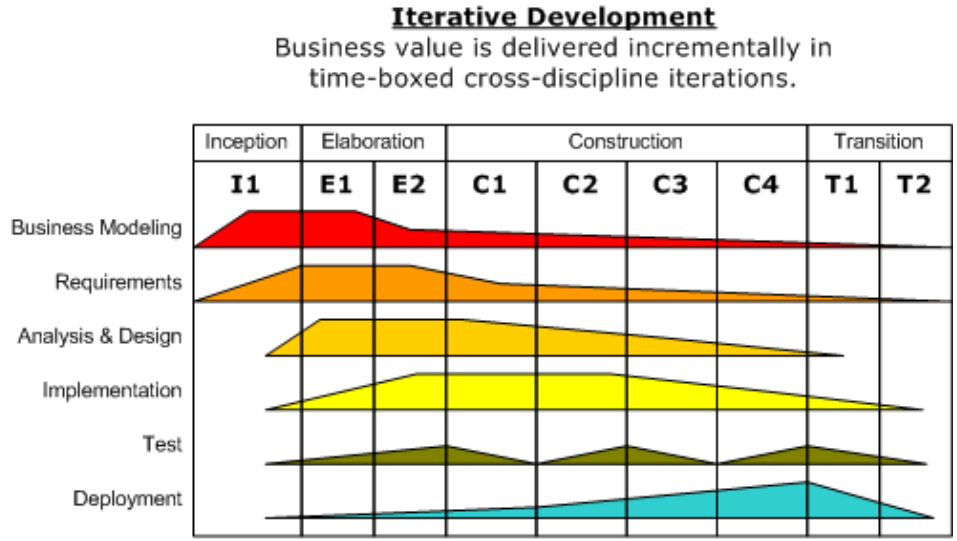
- UP family
  - iterative and incremental development
  - Heavy documentations

Development Methodology

Approaches

- Structured
  - Waterfall development
  - Parallel development
  - V-model
- Rapid application development (RAD)
  - Phased
  - Prototyping
  - Throwaway Prototyping
- Agile development
  - Scrum
  - eXtreme Programming (XP)
  - DSDM
  - DAD
  - …

- UP family
  - iterative and incremental development
  - Heavy documentations



Development Methodology

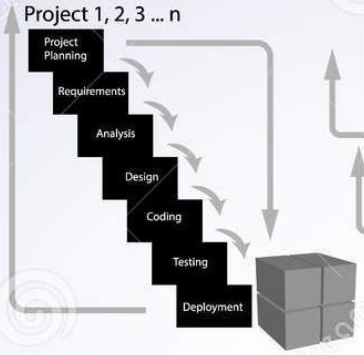
Approaches

- Structured
  - Waterfall development
  - Parallel development
  - V-model
- Rapid application development (RAD)
  - Phased
  - Prototyping
  - Throwaway Prototyping
- Agile development
  - Scrum
  - eXtreme Programming (XP)
  - DSDM
  - DAD
  - …

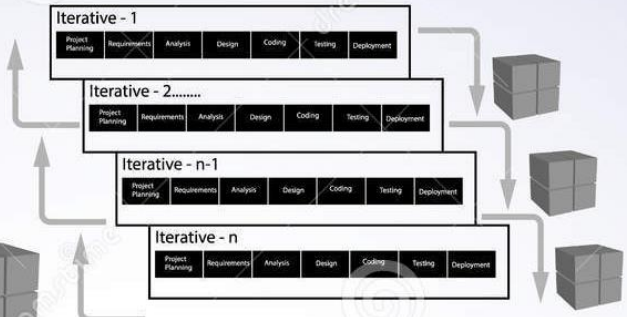
Collection of SDLC Methodologies

Software Development Life Cycle

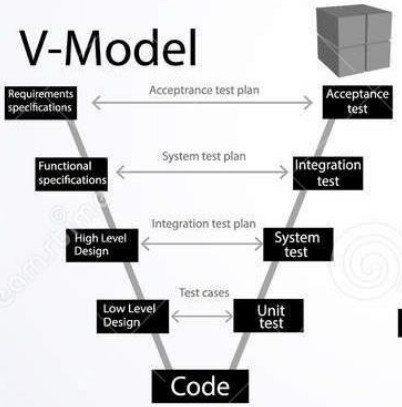
Waterfall



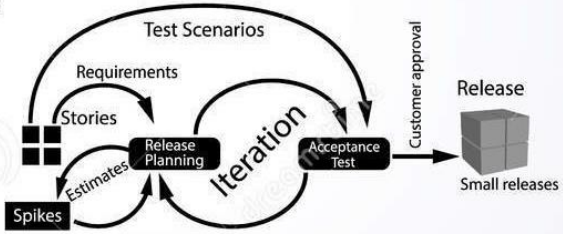
Iterative

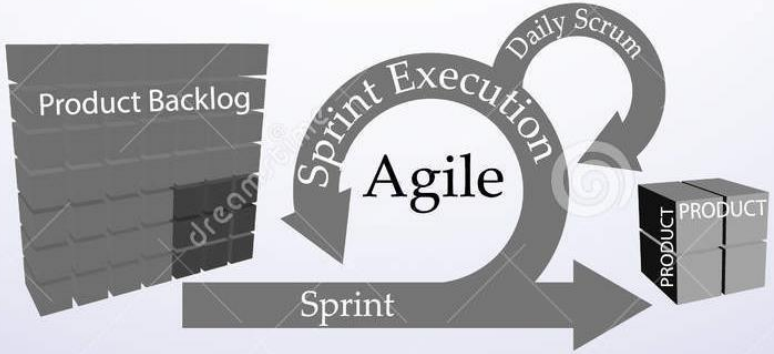


V-Model



Extreme Programming

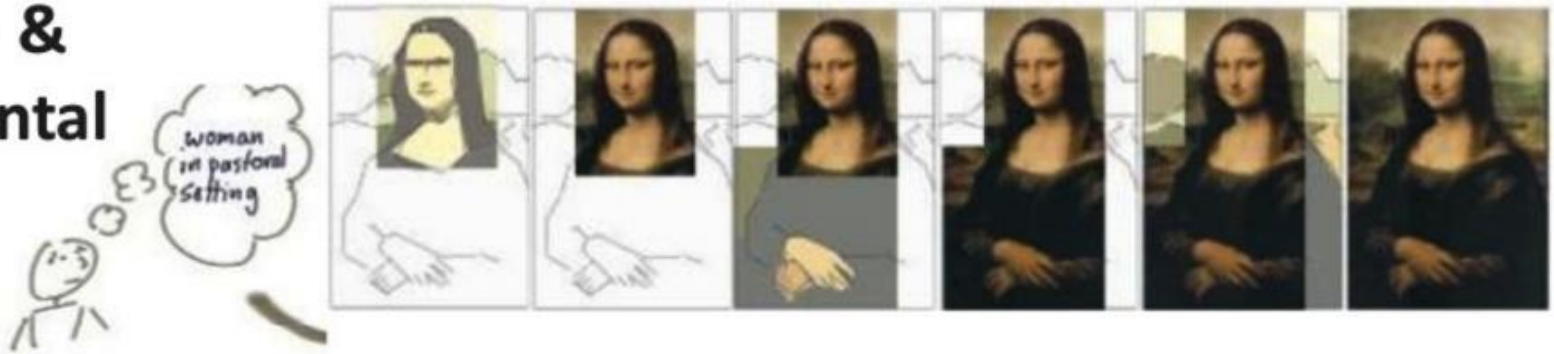


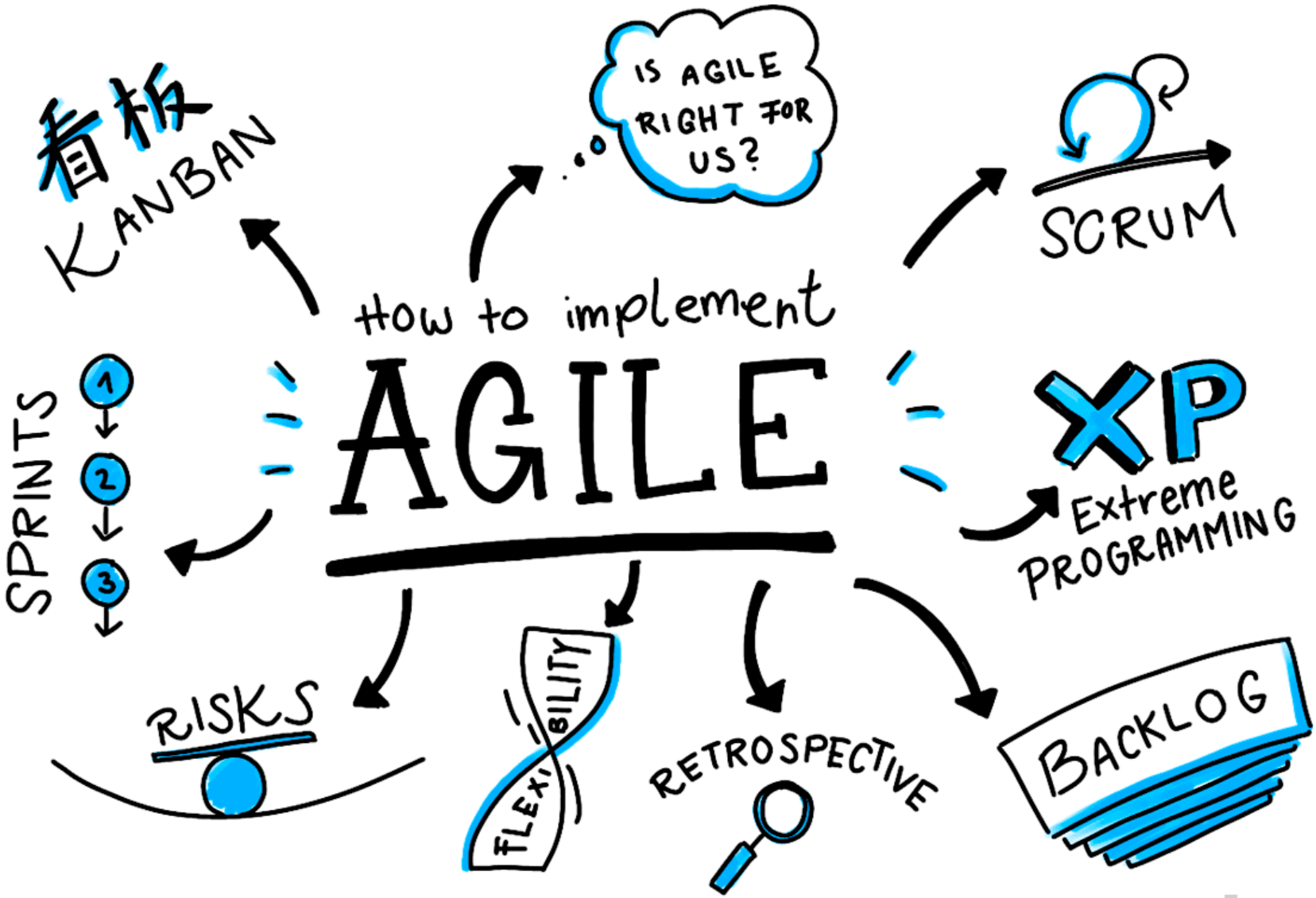


Iterative

Incremental

Iterative & Incremental




看板
KANBAN
IS AGILE RIGHT FOR US?
SCRUM
how to implement
AGILE
SPRINTS
1
2
3
XP
Extreme PROGRAMMING
RISKS
FLEXIBILITY
RETROSPECTIVE
BACKLOG

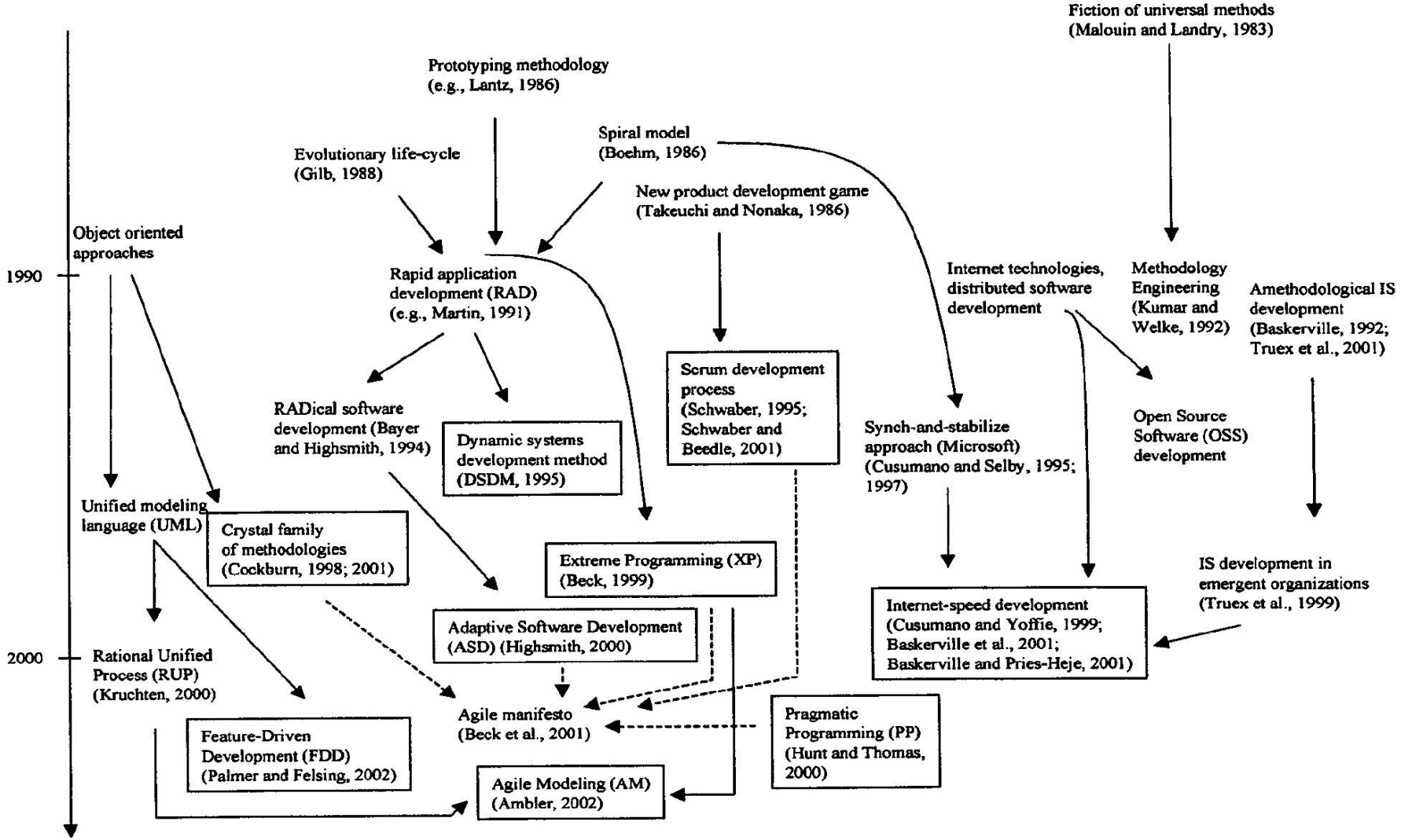
Agile Development - History

- First appeared in 1995.
- The once-common perception that agile methodologies are nothing but controlled code-&-fix approaches, with little or no sign of a clear-cut process, is only true of a small – albeit influential – minority.
- Essentially based on practices of program design, coding and testing that are believed to enhance software development flexibility and productivity.
- Most agile methodologies incorporate explicit processes, although striving to keep them as lightweight as possible.

Major Agile Methodologies

- DSDM – Dynamic Systems Development Method (1995)
- Scrum (1995)
- XP – Extreme Programming (1996)
- ASD – Adaptive Software Development (1997)
- Crystal Family: Orange, Orange Web, Clear (1998, 2001, 2004)
- FDD – Feature-Driven Development (1999)
- AUP – Agile Unified Process (2005)

Agile Methodologies - Evolution Map



[Abrahamsson et al. 2003]

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

"That is, while there is value in the items on the right, we value the items on the left **more**"

Source: http://agilemanifesto.org

Principles of Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Principles of Agile Manifesto

4. people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Principles of Agile Manifesto

7. *Working software* is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

Principles of Agile Manifesto

10. Simplicity--the art of maximizing the amount of work not done--is essential.

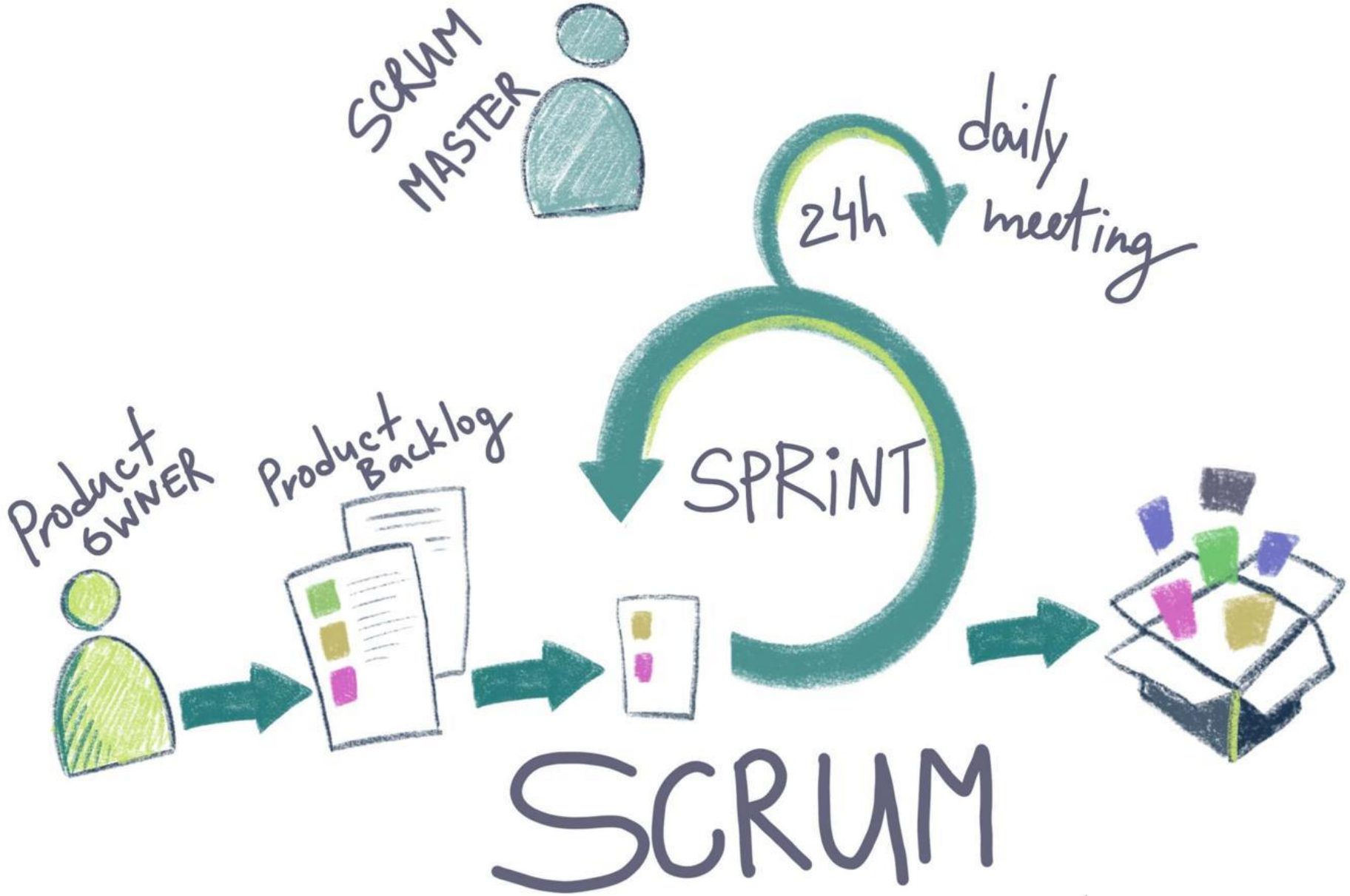
11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: http://agilemanifesto.org

Agile Development

- The Agile way of working breaks a product into functional units according to user stories and prioritizes them to continuously deliver software in short cycles known as iterations.
- Agile is a time-boxed and iterative approach of software delivery.
- It aims to build software incrementally from the start of the project.
- Agile focuses on smaller functional units instead of developing the complete software in a go.
- The Agile way of working largely incorporates feedback loops to help teams to respond to the *ever-changing* outside world.
- The Agile way of working strives towards finalizing a complete task first, before starting with a new one.


SCRUM MASTER
24h
daily meeting
Product OWNER
Product Backlog
SPRINT
SCRUM
@luminousmen.com

Scrum Origins

- First mentioned as a development method in 1986, referring to a fast and flexible product development process practiced in Japanese manufacturing companies (such as Honda and Canon).
- The variant of Scrum used for software development, jointly developed by Sutherland and Schwaber, was introduced in 1995.
- Originally intended as a general framework for systems development, but is currently used as a comprehensive software development methodology.

Scrum.org™

The Home of Scrum

Scrum Origins

- The name emphasizes the importance of teamwork in the methodology and is derived from the game of *rugby*.

Relay race

Scrum

Zombie Scrum

Scrum: New Process Framework

- A people-centric framework based on a set of values, principles, and practices that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.

- Scrum Values: Honesty, Openness, Courage, Respect, Focus, and Collaboration.

- Scrum Principles: Manifestations of the Agile Principles.

- Scrum Practices: Embodied in specific roles, activities, artifacts, and their associated rules.

Scrum: New Process Framework

- A people-centric framework based on a set of values, principles, and practices that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.

- Scrum Values: Honesty, Openness, Courage, Respect, Focus, and Collaboration.

- Scrum Principles: Manifestations of the Agile Principles.

- Scrum Practices: Embodied in specific roles, activities, artifacts, and their associated rules.

Scrum: New Process Framework

- A people-centric framework based on a set of values, principles, and practices that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.

- Scrum Values: Honesty, Openness, Courage, Respect, Focus, and Collaboration.

- Scrum Principles: Manifestations of the Agile Principles.

- Scrum Practices: Embodied in specific roles, activities, artifacts, and their associated rules.

Scrum Values

Scrum: New Process Framework

- A people-centric framework based on a set of values, principles, and practices that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.

- Scrum Values: Honesty, Openness, Courage, Respect, Focus, and Collaboration.

- Scrum Principles: Manifestations of the Agile Principles.

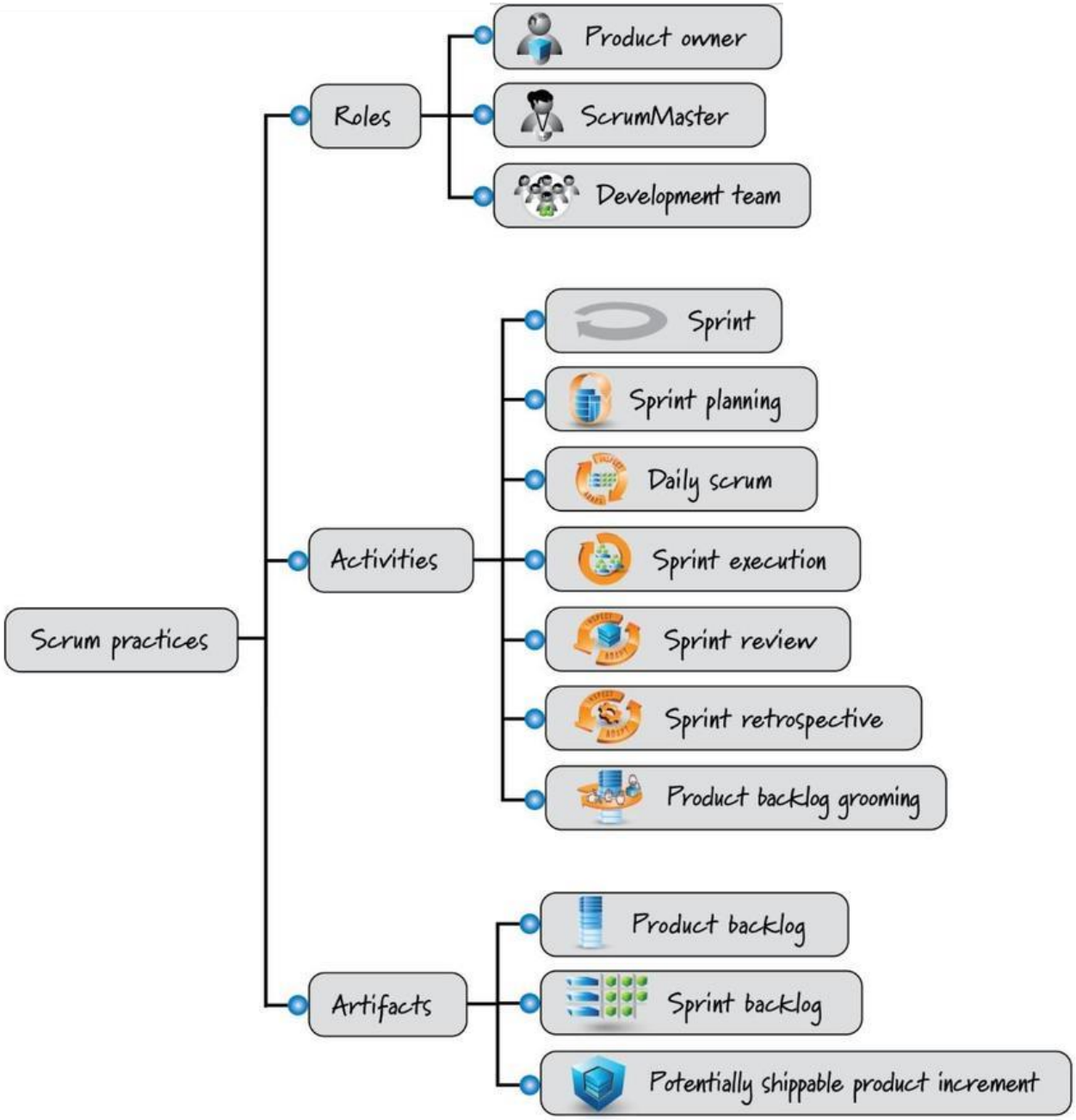
- Scrum Practices: Embodied in specific roles, activities, artifacts, and their associated rules.

Scrum: New Process Framework

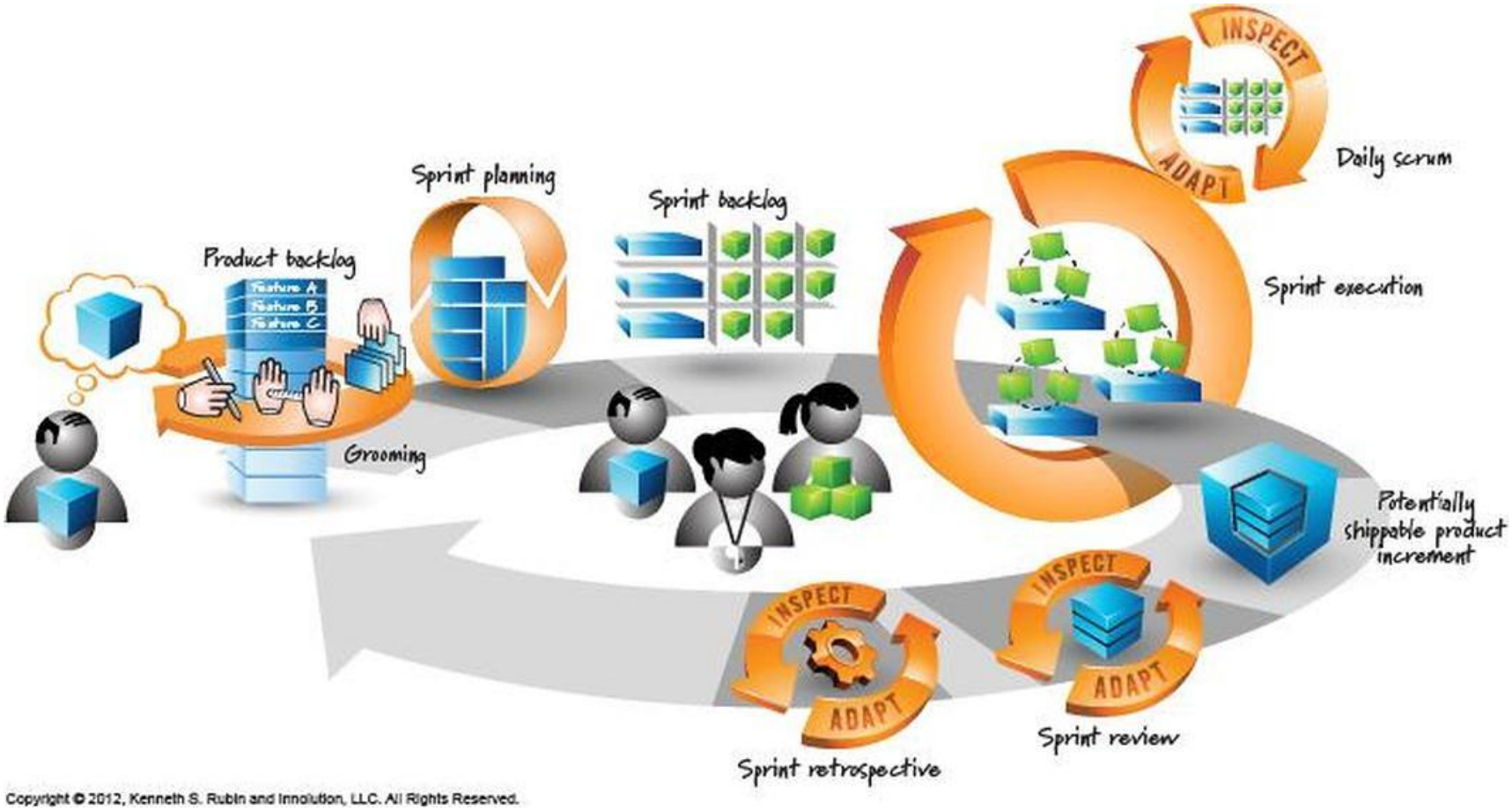
- A people-centric framework based on a set of values, principles, and practices that provide the foundation to which an organization can add its unique implementations for realizing the Scrum practices.
- Scrum Values: Honesty, Openness, Courage, Respect, Focus, and Collaboration.
- Scrum Principles: Manifestations of the Agile Principles.
- Scrum Practices: Embodied in specific roles, activities, artifacts, and their associated rules.

Scrum Practices

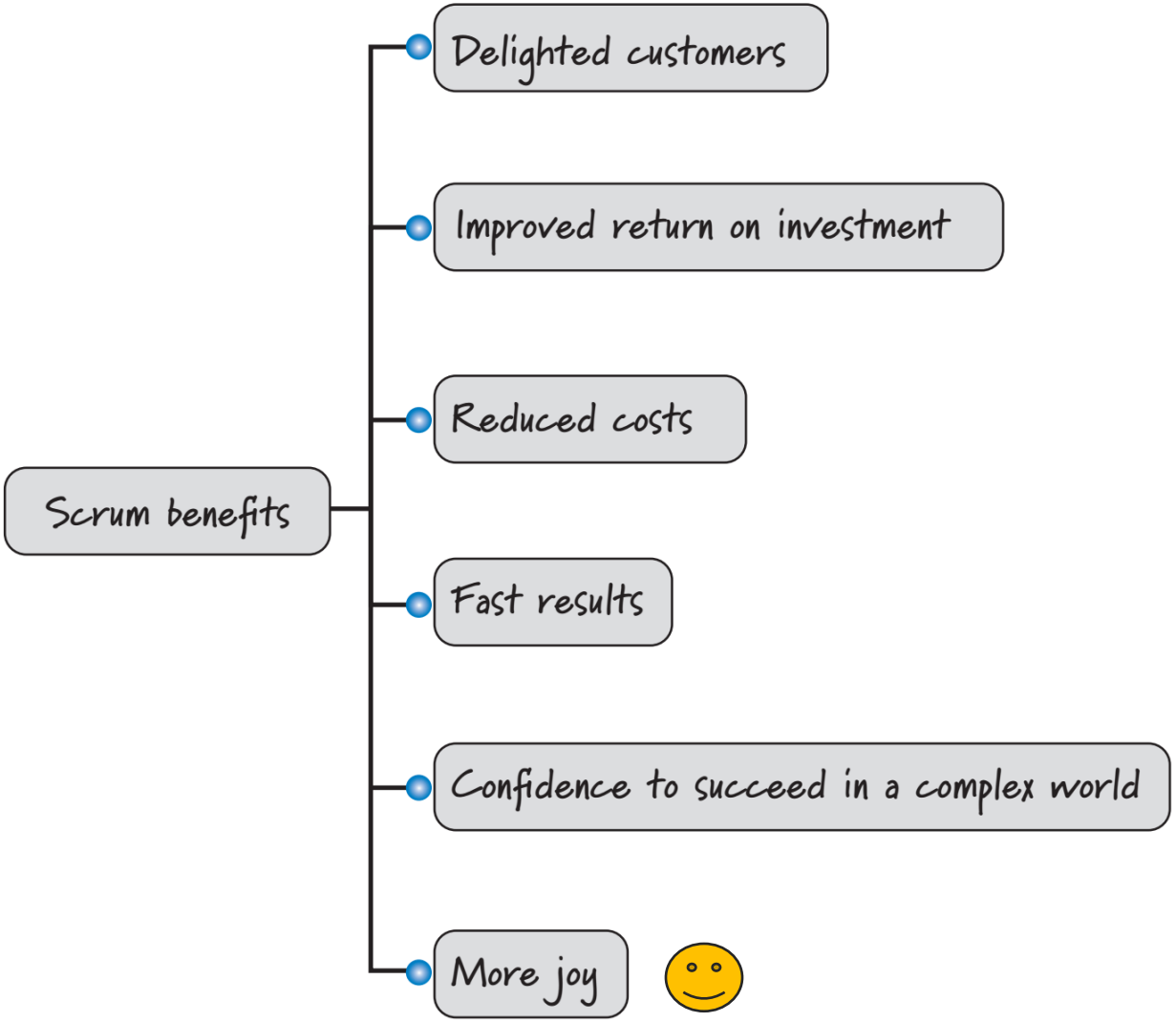
- Roles
- Artifacts
- Activities
- Rules



Scrum – Big Picture



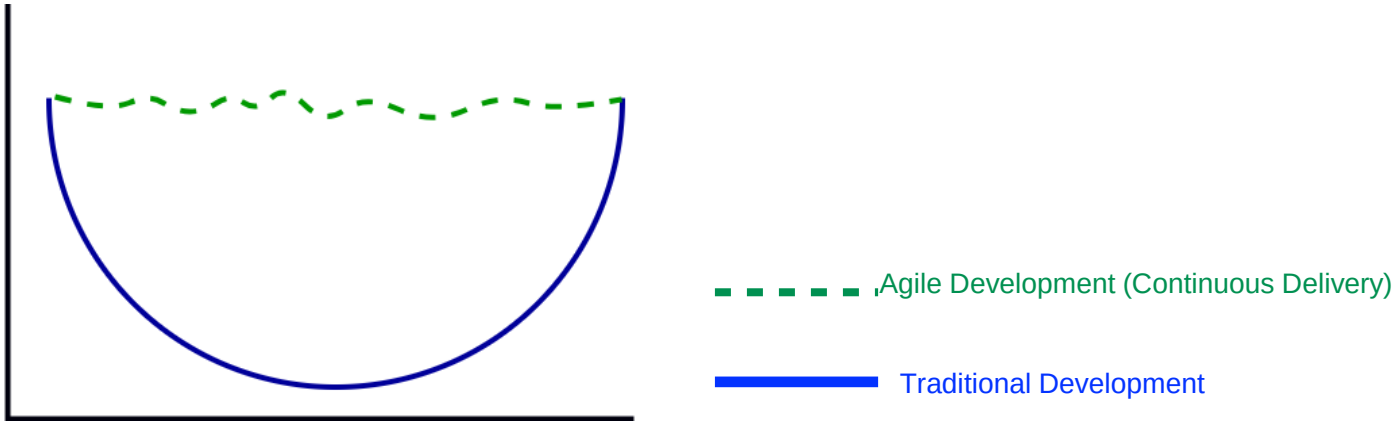
Scrum Benefits



Intangible benefits: Visibility

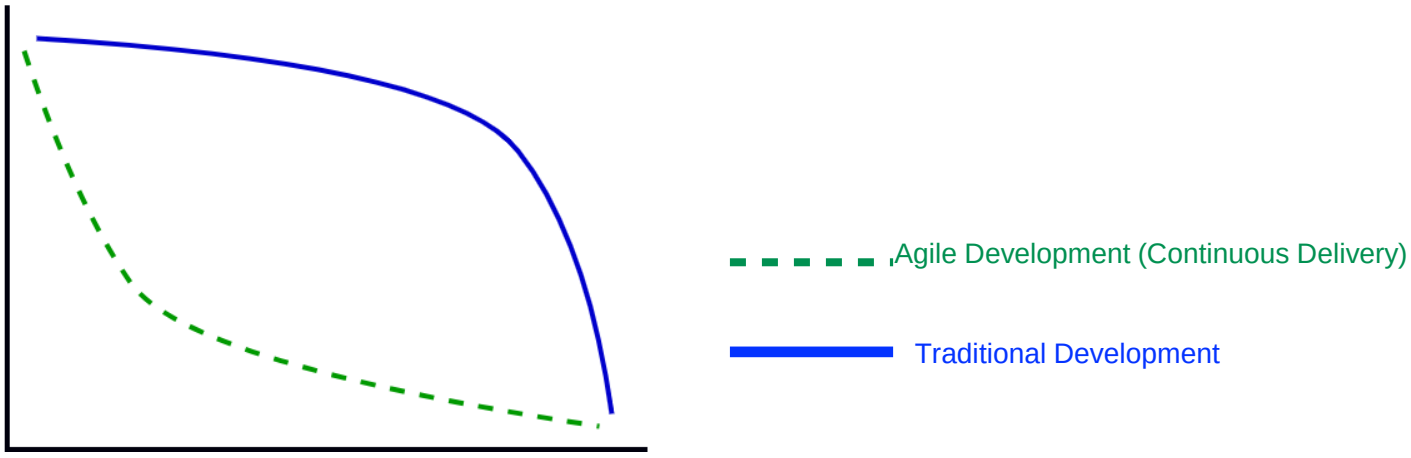
- Product Owner and business are involved with product development on a regular basis

- By attending the Sprintly demo or by launching new shippable features on a regular basis, visibility of what to be delivered is far higher.



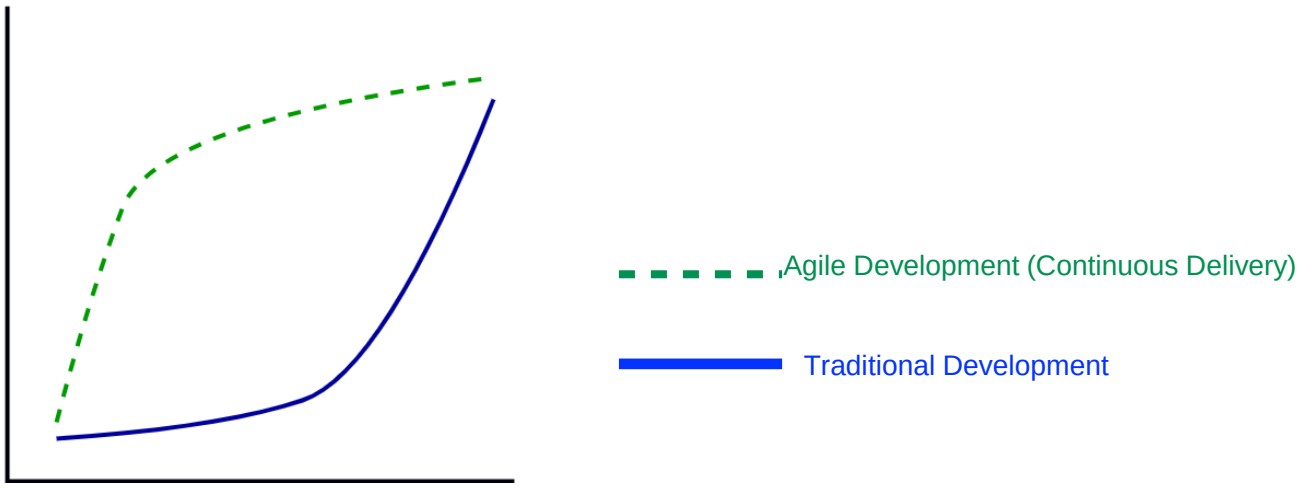
Intangible benefits: Risk

- Optimization of product visibility lowers the risk.
- It becomes clear early in the process whether the team is moving into the right direction and building the right product.
- It is all about feedback and using this feedback to lower risk.



Intangible benefits: Business Value

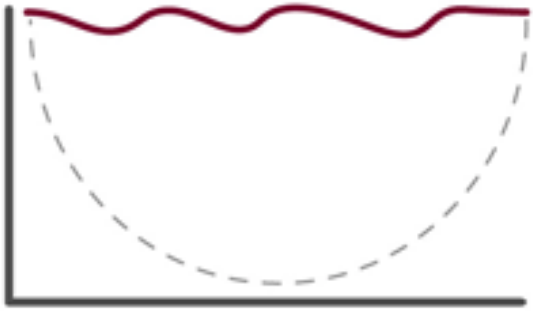
- By delivering a shippable product at the end of each Sprint, the product can be used to generate business value throughout the product development cycle.
- Features are prevented to get ‘stuck’ in the development cycle and are shipped straight away.
- Force the team to consider the valuable feedback of the end-customer through the software development cycle



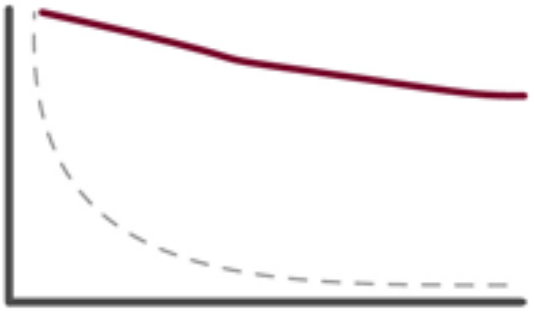
AGILE DEVELOPMENT

VALUE PROPOSITION

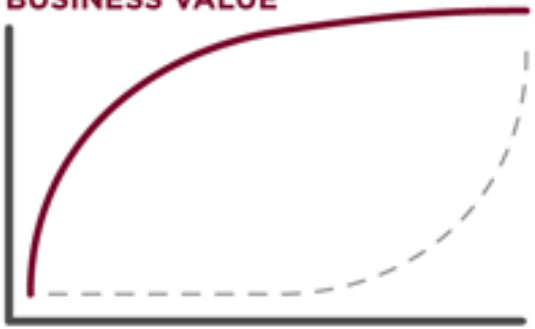
VISIBILITY



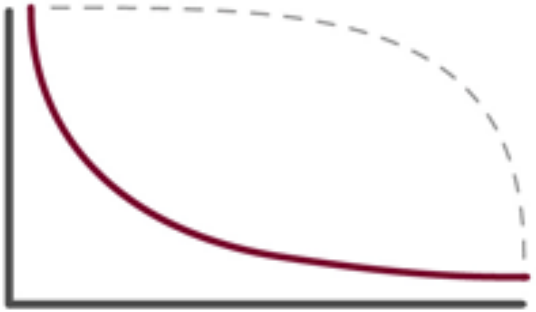
ADAPTABILITY



BUSINESS VALUE



RISK



AGILE DEVELOPMENT — — — TRADITIONAL DEVELOPMENT