Software Deployment

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Table of contents

Software Development Life Cycle

Application Development Life Cycle

- Processes to **design**, **create**, **test**, and **deploy** software.
- Different **methodological models** exist:
  - Waterfall
  - Spiral
  - Agile
  - Incremental
  - Rapid prototyping
  - ...

Main stages

1) Preliminary analysis
2) Design
3) Development
4) Integration and testing
5) Deployment
6) Maintenance
7) End-of-Life

Main stages

1) **Preliminary analysis**
   - Define client's **objectives/problems**
   - Study **existing solutions** (pros and cons)
   - Build **alternative proposals**
   - **Cost benefit** analysis
2) **Design**
3) **Development**
4) **Integration and testing**
5) **Deployment**
6) **Maintenance**
7) **End-of-Life**

Main stages

1) Preliminary analysis
2) Design
    - Define **functions** and **operations**
    - Define **process diagrams**, **class diagrams**, **pseudocode** ...
3) Development
4) Integration and testing
5) Deployment
6) Maintenance
7) End-of-Life

Main stages

1) Preliminary analysis
2) Design
3) Development
    - **Implement** the application
    - Define **unitary tests**
    - Define **documentation**
4) Integration and testing
5) Deployment
6) Maintenance
7) End-of-Life

Main stages

1) Preliminary analysis
2) Design
3) Development
4) Integration and testing
    - **Integrate** all the parts of the application
    - **Test** and **check** bugs
5) Deployment
6) Maintenance
7) End-of-Life

Main stages

1) Preliminary analysis
2) Design
3) Development
4) Integration and testing
5) Deployment
    - **Release** the application
6) Maintenance
7) End-of-Life

Main stages

1) Preliminary analysis
2) Design
3) Development
4) Integration and testing
5) Deployment
6) Maintenance
    - **Support** and **assess** the application
    - **Update** the application
7) End-of-Life

Main stages

1) Preliminary analysis
2) Design
3) Development
4) Integration and testing
5) Deployment
6) Maintenance
7) End-of-Life
    - **Support** is **discontinued**
    - Deprecated product is **no longer sold**
    - **Disposal:** Discontinue the software use $\rightarrow$ **migration**

Software development stages

Software development Stages

0) Pre-alpha
1) Alpha
2) Beta
3) Release Candidate
4) Release To Manufacturing (RTM)
5) General Availability (GA)
6) Production release

Software development Stages

0) **Pre-alpha** Development **steps previous to testing**
   - Requirements analysis
   - Design
   - Implementation
   - Documentation
   - Unitary tests
1) Alpha
2) Beta
3) Release Candidate
4) Release To Manufacturing (RTM)
5) General Availability (GA)
6) Production release

Software development Stages

0) **Pre-alpha**
1) **Alpha**
   First **testing stage** (mostly via white-box techniques)
2) **Beta**
3) **Release Candidate**
4) **Release To Manufacturing (RTM)**
5) **General Availability (GA)**
6) **Production release**

Software development Stages

0) **Pre-alpha**
1) **Alpha**
2) **Beta**
    - **Complete prototype** with still **potential bugs**, released **outside** the development team
    - **Closed beta** *to* **private** | **Open beta** $\rightarrow$ **public**
    - **Perpetual beta** $\rightarrow$ **No final** stable **release**
    - **Goal 1: Detect bugs** (Beta testers $\rightarrow$ issues reporting)
    - **Goal 2: Demonstrate** the **product**
3) **Release Candidate**
4) **Release To Manufacturing (RTM)**
5) **General Availability (GA)**
6) **Production release**

Software development Stages

0) **Pre-alpha**
1) **Alpha**
2) **Beta**
3) **Release Candidate**
    - a.k.a. **Going silver**
    - **Beta** with **potential** to become a **final product**.
    - All features **designed**, **implemented** and **tested**.
    - **No significant bugs**
4) **Release To Manufacturing (RTM)**
5) **General Availability (GA)**
6) **Production release**

Software development Stages

0) **Pre-alpha**
1) **Alpha**
2) **Beta**
3) **Release Candidate**
4) **Release To Manufacturing (RTM)**
    - a.k.a. **Going gold** or **Release to Marketing**
    - **Digitally signed**: to **confirm software author** and **guarantee** that it is **not altered**)
    - **Stable release** with **sufficient quality** to undergo **mass distribution**
5) **General Availability (GA)**
6) **Production release**

Software development Stages

0) Pre-alpha
1) Alpha
2) Beta
3) Release Candidate
4) Release To Manufacturing (RTM)
5) General Availability (GA)
    - a.k.a. **Gold**
    - **Commercialization** steps: **completed**
6) Production release

Software development Stages

0) Pre-alpha
1) Alpha
2) Beta
3) Release Candidate
4) Release To Manufacturing (RTM)
5) General Availability (GA)
6) Production release
    - **Physical** media
    - Release to **web**

Project Management Models

Main Project Management Elements

- **Requirements**: Conditions/tasks that should be achieved for a successful project completion.
- **Documentation**: Official information that serves as record
- **Deliverable**: Main object to be provided as part of a project product.

Two important Project Management Models: **Waterfall** and **Agile**

Waterfall

Requirements Analysis

↓

Software **Design**

↓

Software **Implementation**

↓

Verification

↓

Deployment

↓

Maintenance

- **Sequential** approach
- **Traditional** approach
- **Criticized** as being too **inflexible**

Four Values of the Agile Manifesto

`https://agilemanifesto.org/principles.html`

- **Individuals and interactions** *over process and tools* [Team dynamic]
- **Working software** *over comprehensive documentation* [value Delivery]
- **Customer collaboration** *over contract negotiation* [Business collaboration]
- **Responding to change** *over following a plan* [Retrospectives]

Agile

- **Software** and **Requirements** co-evolve through the **interaction** with the **client**.
- **Adaptive planning**, **Continual improvement**, **evolving development**
- **Short feedback loops**, **adaptation cycles**
- **Frameworks**:
  - Scrum (58%), ScrumBan (10%), Scrum/XP Hybrid (8%)
  - Kanban (7%)
  - Iterative Development (4%)
  - Other (13 %)

Agile | Kanban

- **Visual feedback** about the work in progress status
- Project team accepts only a sustainable amount of **Work-In-Progress** (WIP)
- **Flow**: Focus on less work at a time $\rightarrow$ improve efficiency

Agile | XP: extreme programming

Take good development practices to extreme levels:

- **Design**: describe code/behaviour (simple $\rightarrow$ complex)
- **Code**: Clear, readable and concise code
- **Test**: Check all possible features before continuing
- **Listening**: Product should match requirements

Innovative practices:

- Pair-programming
- Continuous integration and Refactoring
- Design continuous improvement (start simple)
- Write tests not requirements.

Agile | Scrum

”Framework for developing, delivering, and sustaining complex products”

- **Iterative** approach
- **Incremental** approach
- **Empiricism**-based philosophy

3 Scrum Pillars:

- Transparency
- Inspection
- Adaptation

5 Scrum Values:

Commitment, Courage, Focus, Openness, Respect

Motivate the team by providing:

- **Mission**: short statement constant **Why?**
- **Product vision**: responsibilities and limits **What the work will be like?**

Agile | Scrum

- **Sprint**:
  Predefined **basic development unit** (7d $\rightarrow$ 30d).
- **Sprint planning event**:
  Set Prod. backlog (requirements, work to do, forecast)
- **Sprint review** and **print retrospective**:
  - Choose progress to show (potentially **releasable**)
  - Identify possible **improvements**
- **Daily scrum**:
  - Daily stand-up meeting **limited to 15min**
  - Starts precisely **on time**, **same hour**, **same place**
  - Team members **come prepared**:
    - What did I **complete yesterday**?
    - What do I plan to **complete today**?
    - Are they **impediments/risks/issues**?

Agile | Scrum

Development Team (3→ 9 people):

- Responsible for delivering product
- Cross-functional, self-organizing, supportive

Product Owner:

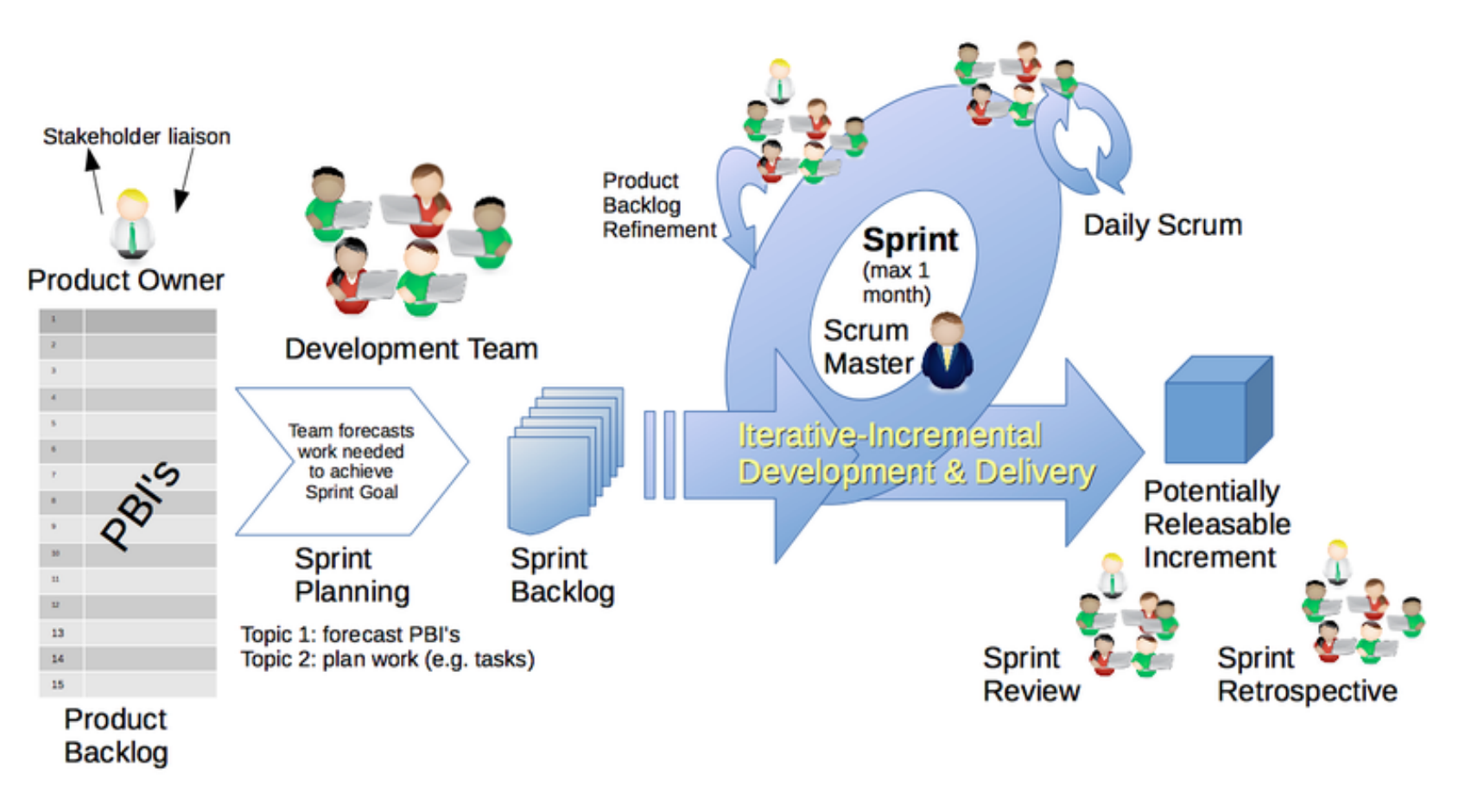
- Maximize **product value**, ensure **right** product is built
- Inventory/prioritize the work (customer-focused)
- Decisive, flexible, collaborative, optimistic, available

Scrum Master:

- **Facilitate team Productivity** through **Agile principles** and **agreed processes/practices**
- **Share/exchange information** from/to outside
- Organized, supportive, communicative leaders

Agile | Scrum

https://scrumguides.org/



Choosing management approach: VUCA

- **V**: Volatility i.e., rate of change for a situation
- **U**: Uncertainty i.e., lack of predictability.
- **C**: Complexity i.e., large number of interacting elements influencing the project.
- **A**: Ambiguity i.e., possible misunderstanding of causes of events or circumstances.

High VUCA $\rightarrow$ Agile

Low VUCA $\rightarrow$ Waterfall

Blended approaches are also possible

Conclusion