Software Deployment

Reminders on Software Development

Sergio Peignier   sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

- **Software**: Set of **computer instructions** defining what the **computer** should **do**.

- **Hardware**: **Physical components** of a computer.

Software **Development** $\rightarrow$ Software **Deployment**

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
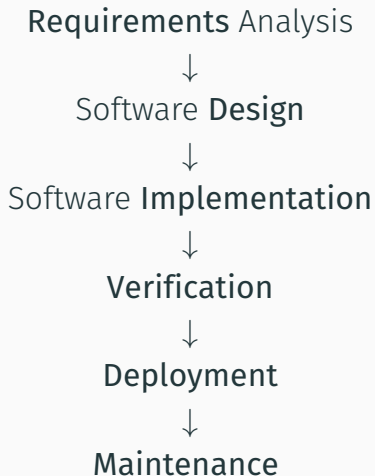6) Maintenance
7) End-of-Life

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
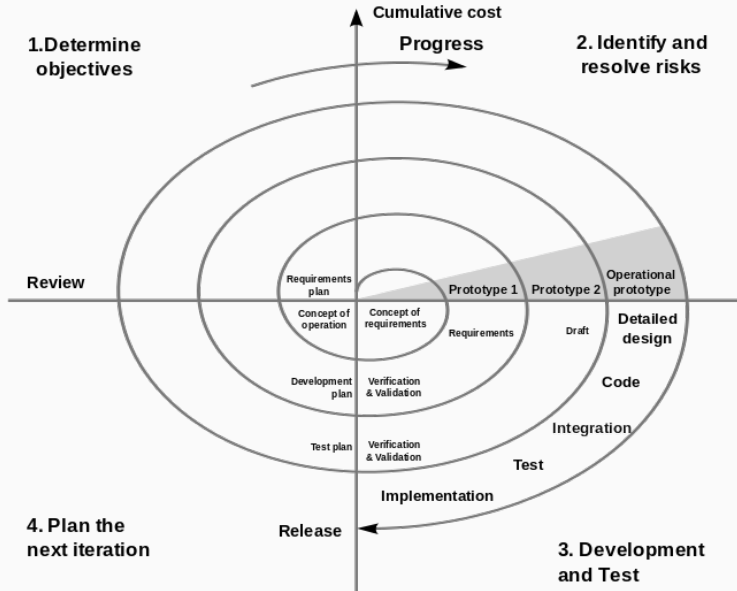   - Disposal: Discontinue the software use $\rightarrow$ migration

Some Software Development Life Cycle Models

Waterfall

Requirements Analysis
↓
Software Design
↓
Software Implementation
↓
Verification
↓
Deployment
↓
Maintenance

- Sequential approach

- Traditional approach

- Criticized as being too inflexible

Spiral Development

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, Kanan, …

Agile | Scrum

- **Sprint**:
  Predefined **basic development unit** (7d $\rightarrow$ 30d).
- **Sprint planning event**:
  Set 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
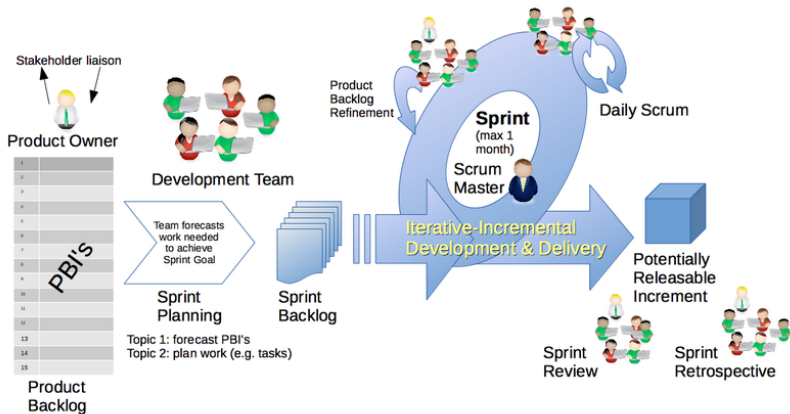
Software development stages

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 → public
   - Perpetual beta → No final stable release
   - Goal 1: Detect bugs (Beta testers → 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**

Conclusion

Software Development Overview (using carrot2)