

02-SOFTWARE PROCESS AND AGILE SOFTWARE DEVELOPMENT

Royana Afwani
PSTI FT Universitas Mataram
2020

OVERVIEW:

1. Apa yang dimaksud dengan *software process*?
2. Apa saja kerangka kerja yang umum ada dalam sebuah *software process*?
3. Apa itu *software process flow*?
4. Mengapa kelincahan (*agile*) sangat diperlukan dalam pekerjaan merekayasa *software*?
5. Apa itu *Agile Software Development*, dan apa perbedaannya dengan metode pengembangan *software* tradisional?

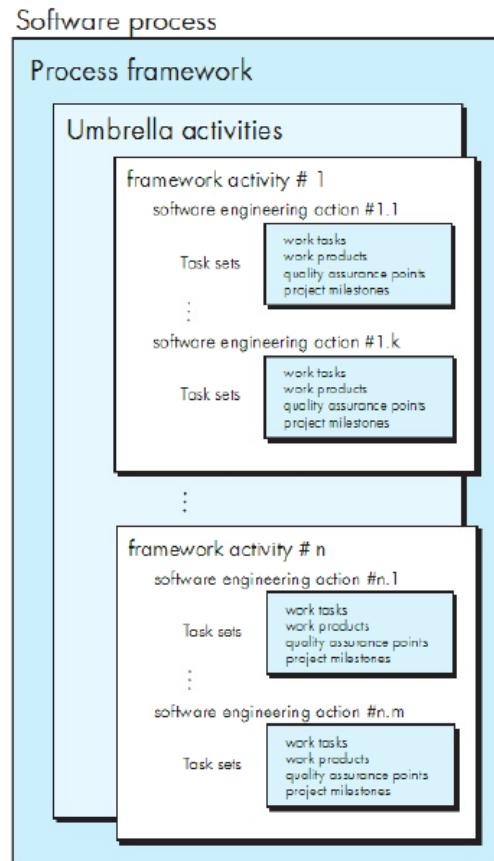
A SOCIAL LEARNING PROCESS

“Karena software, seperti semua modal lainnya, dibangun dengan pengetahuan, dan karena pengetahuan tersebut bersifat tersebar, susah diungkapkan, terpendam dan tidak komplit dalam cakupan yang luas, maka pengembangan perangkat lunak (*software development*) adalah sebuah proses pembelajaran sosial”.

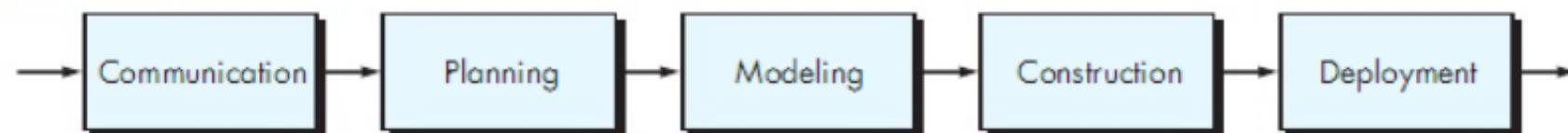
A GENERIC PROCESS FRAMEWORK:

- Proses adalah sekumpulan aktifitas, aksi dan tugas yang dilaksanakan ketika sebuah produk akan dibuat.
- Kerangka kerja yang umum **Generic Framework**: **communication** (komunikasi) , **planning** (perencanaan), **modelling** (pemodelan), **construction** (konstruksi), dan **deployment** (penyebaran).
- Sebagai tambahan aktifitas seperti: *project tracking and control, risk management, quality assurance, configuration management, technical reviews* dan lainnya secara iterative (umbrella) ditambahkan kedalam prosesnya.
- **Process flow** (aliran proses): aktifitas, aksi, dan tugas-tugas diorganisir berdasarkan urutannya.

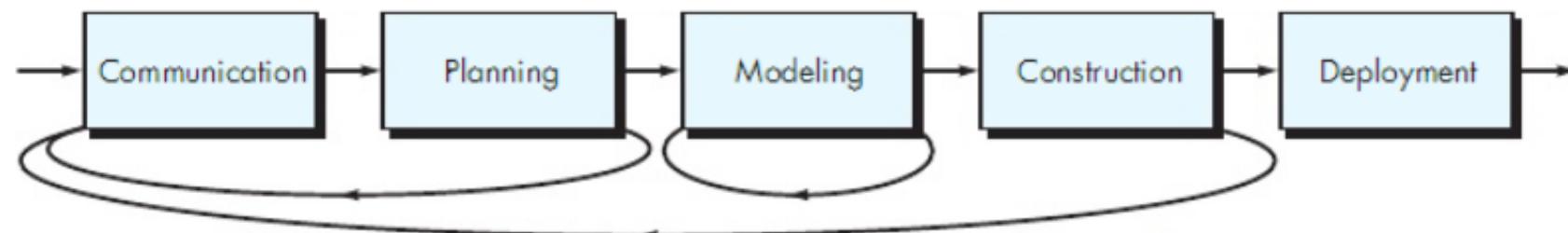
SOFTWARE PROCESS FRAMEWORK:



PROCESS FLOW

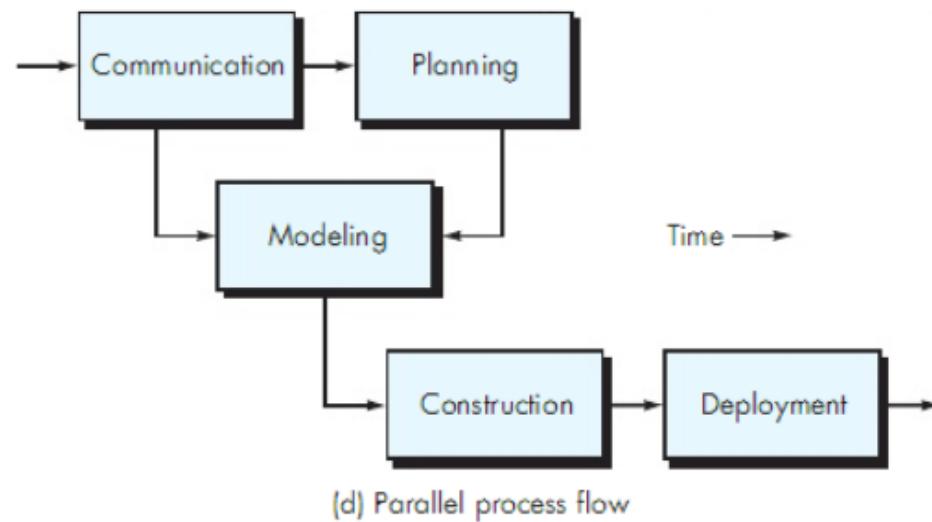
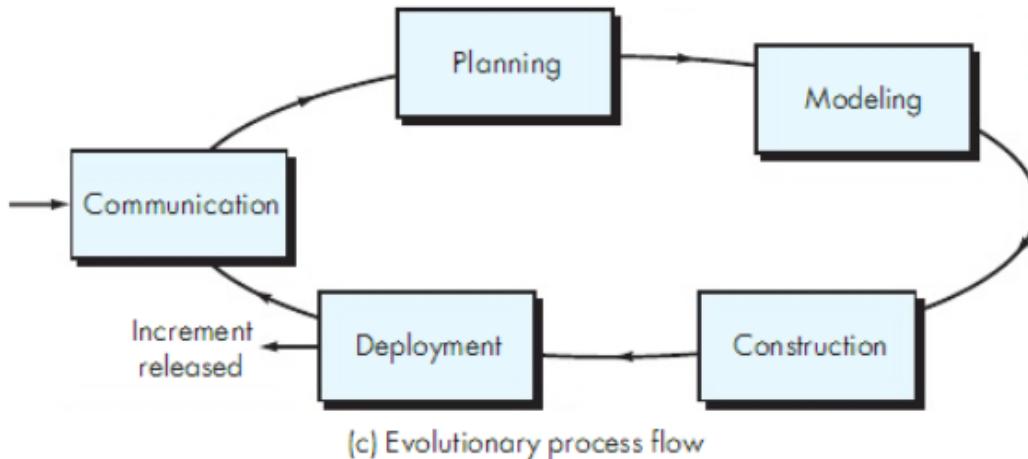


(a) Linear process flow

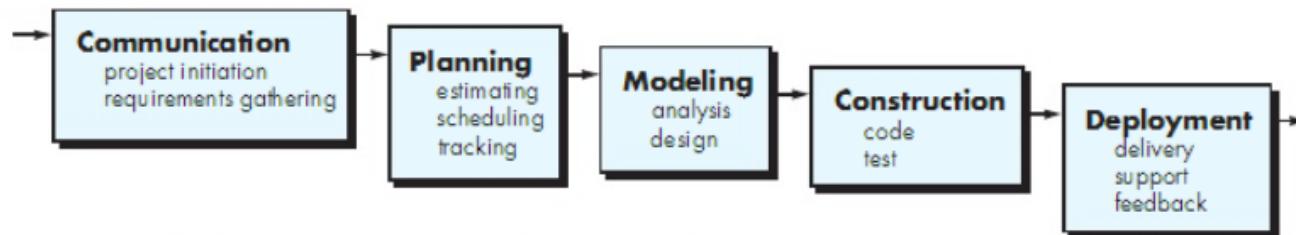


(b) Iterative process flow

PROCESS FLOW:



PROCESS MODELS: WATERFALL

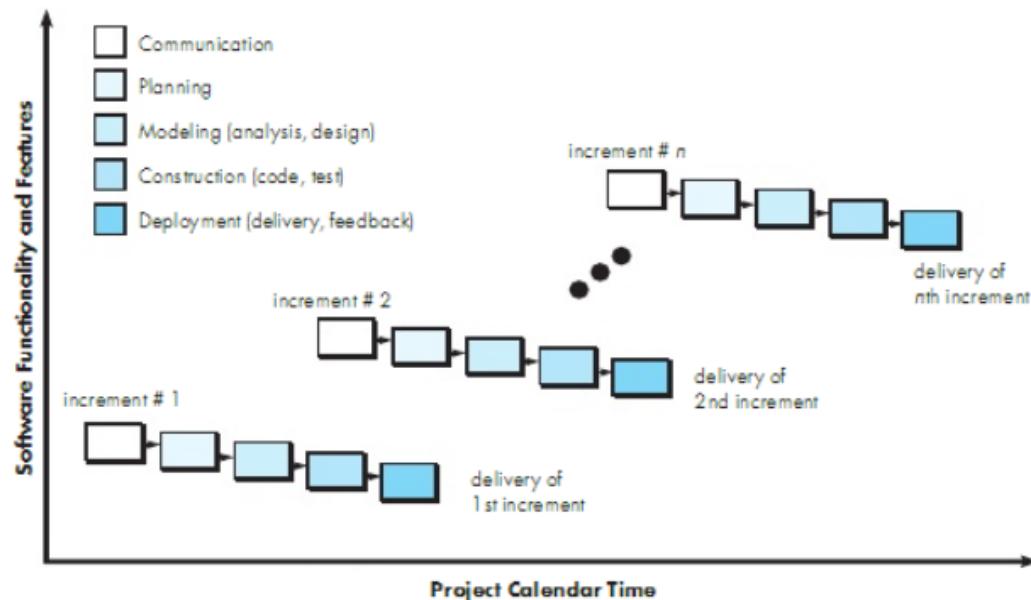


- Waterfall, sering disebut dengan *Classic Life Cycle*. Menggambarkan proses pengembangan perangkat lunak dengan pendekatan yang sistematik dan sekuensial (berurutan).

KELEMAHAN MODEL WATERFALL:

1. Pada kenyataannya sangat jarang sebuah proyek mengikuti pola aliran sekuensial. Walaupun Waterfall mendukung iterasi, namun secara tidak langsung. Perubahan akan sulit diaplikasikan.
2. Biasanya sangat sulit bagi pelanggan untuk memberikan *requirement* secara eksplisit. Umumnya pernyataan tentang keinginan pelanggan dibahasakan secara ambigu.
3. Pelanggan harus bersabar. *Release* baru dapat dilakukan saat keseluruhan sekuens berakhir.

PROCESS MODEL: INCREMENTAL

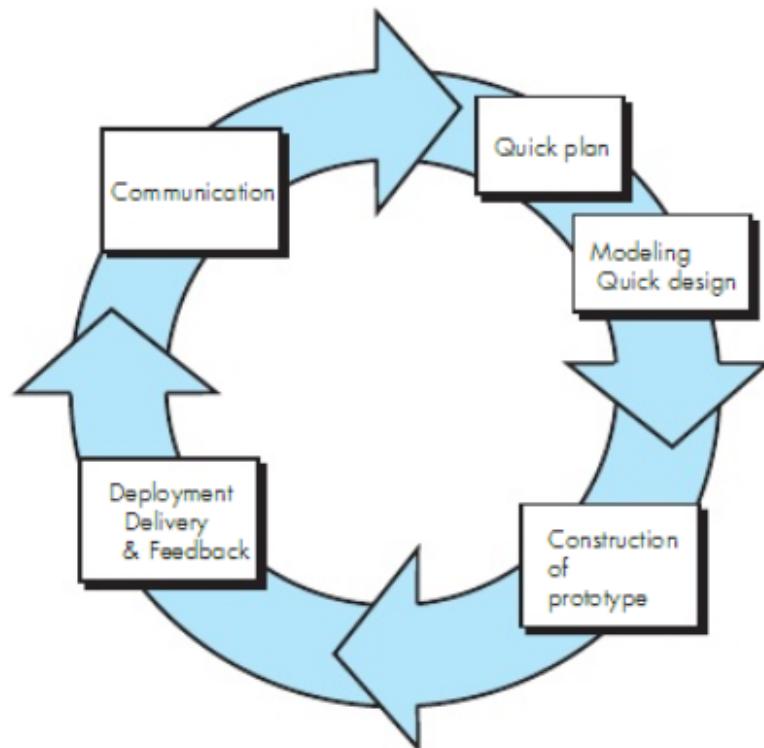


- Proses incremental adalah kombinasi dari proses linear dan parallel.
- Setelah pemodelan umumnya tim dapat menggambarkan kembali fitur berikutnya yang akan dikembangkan, sehingga proses incremental berikutnya dapat dilaksanakan.

PROCESS MODEL: EVOLUTIONARY

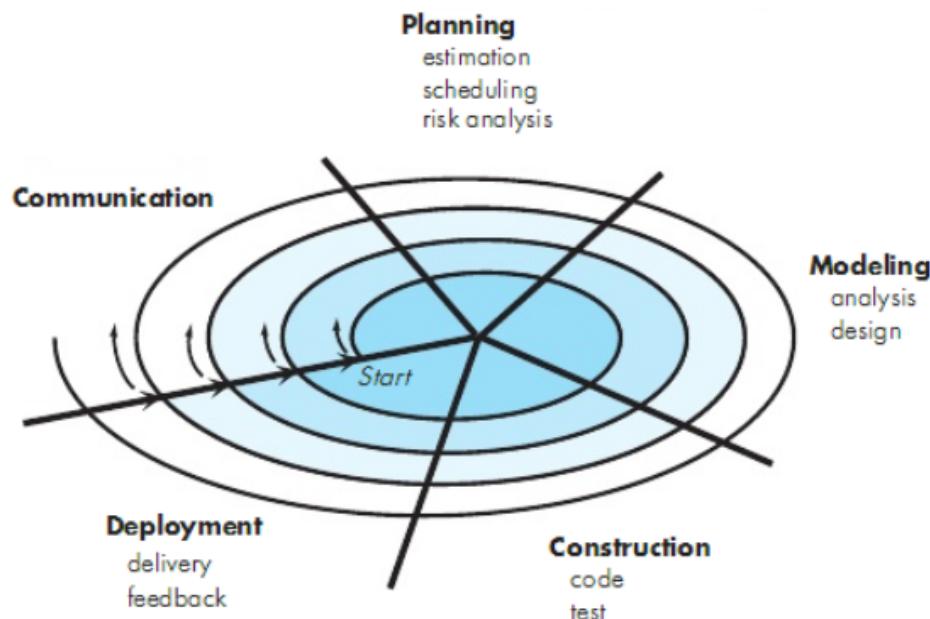
- Software, seperti sistem kompleks lainnya, berevolusi seiring jalannya waktu.
- Model Evolutionary bersifat iteratif.
- Bertujuan untuk memberikan kemampuan kepada developer untuk membangun software secara berkesinambungan, sehingga software yang lebih lengkap dapat dibangun dari waktu ke waktu.
- Bentuk model Evolutionary yang umum:
 - Prototyping
 - Spiral

EVOLUTIONARY: PROTOTYPING



- Mendukung requirement yang kurang detail, akan lebih didetaillkan pada proses incremental berikutnya.
- Mendukung ketika developer kurang yakin tentang apa yang akan mereka hadapi.
- Baik developer maupun pelanggan menyukai metode ini.
- *Bila demikian, coba Anda identifikasi apa yang menjadi kekurangan model ini?*

EVOLUTIONARY: SPIRAL



- Dengan model Spiral, software dibangun dalam serangkaian *release* secara *revolutionary*.
- Pada release awal, software mungkin berbentuk prototype, namun akan berkembang menjadi proses yang lebih lengkap pada incremental berikutnya.

TRADITIONAL SOFTWARE DEVELOPMENT METHODS ?

The problem of the traditional software development methods that they need a very heavy process, such as [1]:

1. The project's plan must be done in advance.
2. The written software's requirements.
3. Fully design that satisfy the written requirements.
4. Build the code of the software that satisfies all the written requirements and the designs.
5. Fully test the software and check if it satisfies the requirements and the design.

Many projects that follow the traditional software development methods addressed major problems, especially in the maintenance and the changes based on the user's requests. Some of these changes may lead to major changes which are considered as a big problem in software development. Because all of that the need for a lightweight software development method is needed, the main goal of these methods is to speed up the development and effectively respond to the requested changes. This lightweight software development method is called Agile software development methods. [1].

AGILE DEVELOPMENT

Agile Software Development manifesto [2] :

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
- Customers collaboration over contract negotiation
- Responding to change over following a plan

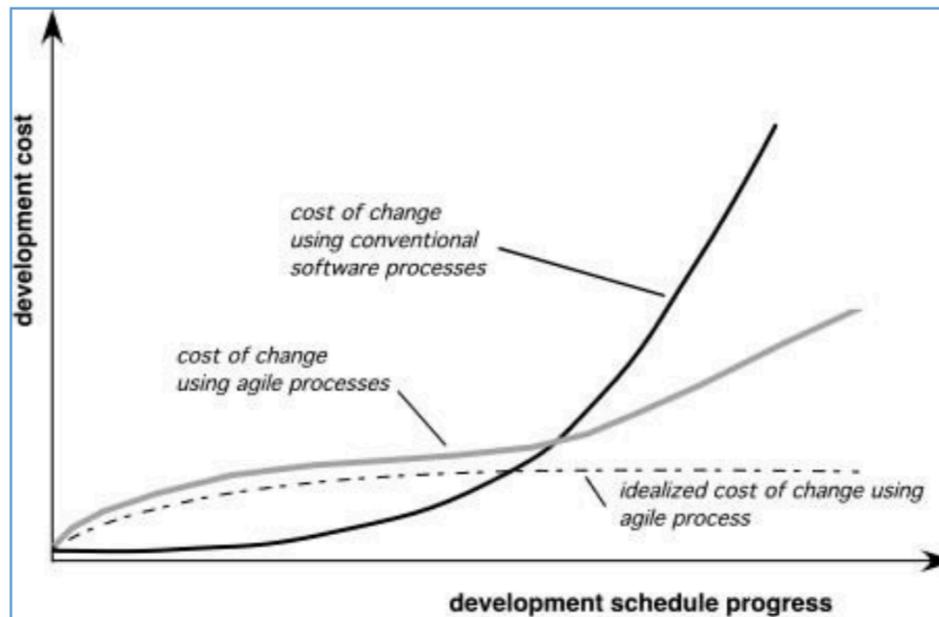
[2] Al-Saqqa, Samar, Samer Sawalha, and Hiba AbdelNabi. "Agile Software Development: Methodologies and Trends." *International Journal of Interactive Mobile Technologies* 14.11 (2020).

AGILITY PRINCIPLE [3]:

1. Prioritas yang paling tinggi adalah untuk memuaskan pelanggan dengan menyediakan software bermutu dengan cepat dan berkala.
2. Mampu menangani perubahan requirement bahkan ketika sudah dalam tahap pengembangan.
3. Me-release working software secara berkala.
4. Orang bisnis dan developer bekerja sama setiap hari selama proyek berlangsung.
5. Membangun proyek dalam lingkungan individu yang termotivasi.
6. Metode yang paling tepat untuk berkomunikasi antar individu adalah face-to-face.
7. Working software adalah ukuran utama kemajuan proyek.
8. Agile process mempromosikan pengembangan yang berkelanjutan.
9. Perhatian yang terus menerus terhadap kesempurnaan teknik dan desain.
10. Kesederhanaan adalah hal yang sangat penting.
11. Arsitektur, requirement dan desain yang terbaik dating dari tim yang terbentuk dengan sendirinya.
12. Pada interval berkala, tim merefleksikan dirinya untuk menjadi lebih efektif dan lebih baik lagi.

[3] Conboy, K. (2009). "Agility from first principles: Reconstructing the concept of agility in information systems development". *Information systems research*, 20(3), 329-354. <https://doi.org/10.1287/isre.1090.0236>

THE DEV. COST VS CHANGE IN THE DEVELOPMENT PROCESS



Ref : Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.

AGILE DEVELOPMENT VS TRADITIONAL DEVELOPMENT

Parameter	Traditional Methods	Agile Methods
Ease of Modification	Hard	Easy
Development Approach	Predictive	Adaptive
Development Orientation	Process Oriented	Customer Oriented
Project Size	Large	Small or Medium
Planning Scale	Long Term	Short Term
Management Style	Command and Control	Leadership and Collaboration
Learning	Continuous Learning while Development	Learning is secondary to Development
Documentation	High	Low
Organization Type	High Revenue	Moderate and low Revenue
Organization's Number of Employees	Large	Small
Budget	High	Low
Number of Teams	Multiple	One
Team Size	Medium	Small

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). "Empirical study of agile software development methodologies: A comparative analysis". ACM SIGSOFT Software Engineering Notes, 40(1), 1-6. <https://doi.org/10.1145/2693208.2693233>

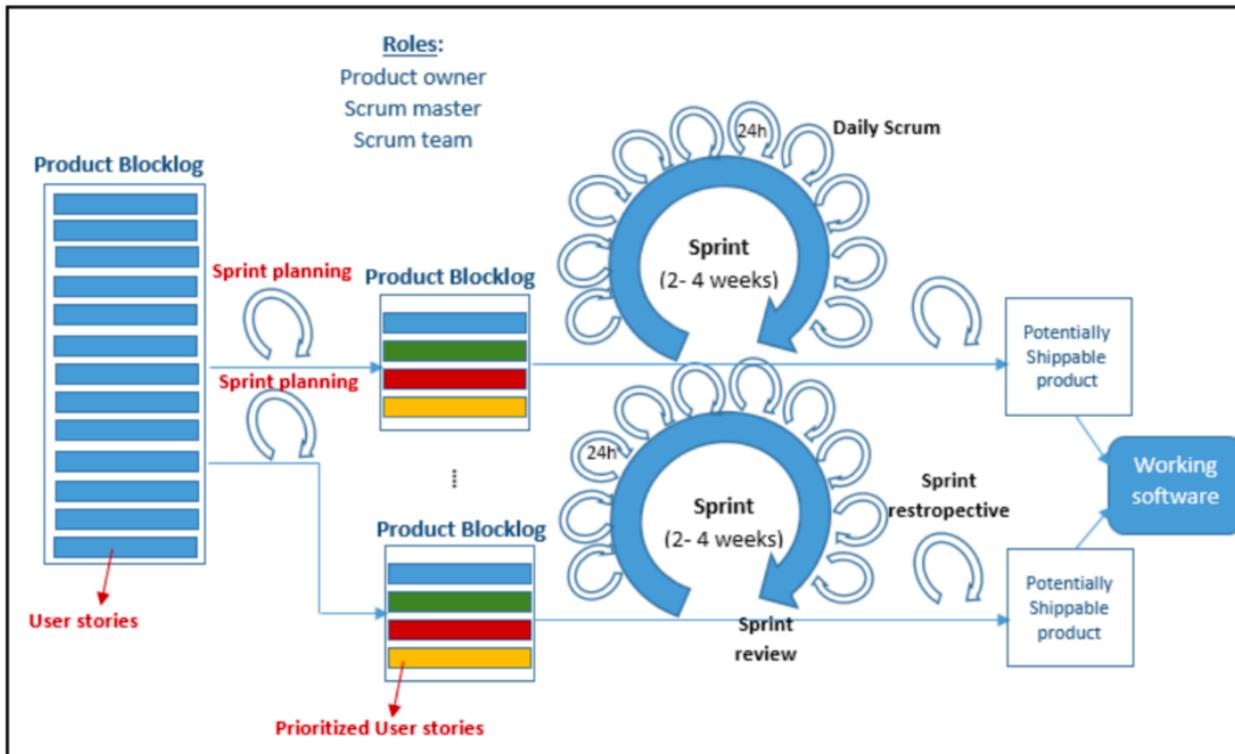
Vijayasarathy, L. R., & Butler, C. W. (2016). "Choice of software development methodologies: Do organizational, project, and team characteristics matter?". IEEE software, 33(5), 86-94. <https://doi.org/10.1109/ms.2015.26>

AGILE METHOD

Test-Driven Development (TDD) method,
Feature Driven Development (FDD) method,
Extreme Programming (XP) method,
Scrum method,
Dynamic System Development Model (DSDM) method and
Crystal methods etc.

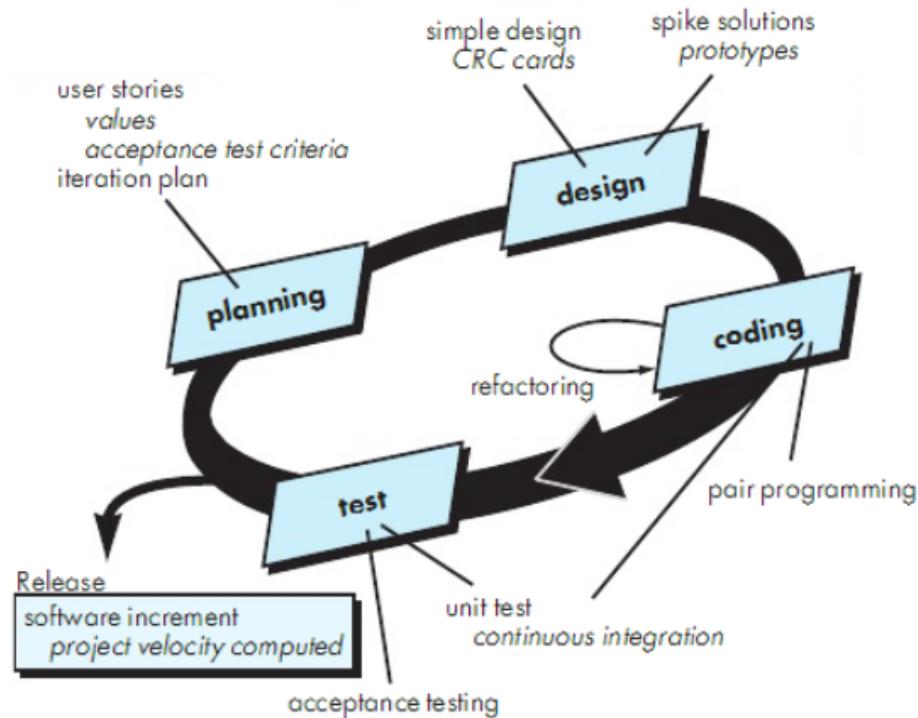
Each method has its own principles, life cycle, roles, advantages and disadvantages etc

SCRUM METHOD LIFE CYCLE



Braude, E. J., & Bernstein, M. E. (2016). Software engineering: modern approaches. Waveland Press.

EXTREME PROGRAMMING (XP):



- XP menggunakan pendekatan berorientasi obyek sebagai paradigma yang disarankan.
- Aktifitas dari kerangka kerjanya: planning, design, coding, dan testing.

BELAJAR SAMA SAMA:

Masing-masing kelompok (2 orang) membuat key presentasi (PPT) , memilih salah satu dari banyak Agile System Development.

Key Presentasi minimal terdapat : Filsosofi/Makna, Proses/Tahapannya, Kelebihan, Kekurangan, Kesimpulan.

Ketentuan : Bahasa (Jelas/padat dan EYD benar), terdapat minimal 2 paper referensi.

Pengumpulan liat assignment pada berajah.