

Unit 1(Part 2)

Agile Development

Prepared by :
Dr. Pooja M Bhatt
Computer Engineering Department,
MBIT

Index

- Agility & agile process model
- Extreme programming(XP)
- Other Process Model of Agile Development and Tools
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Feature Driven Development (FDD)

Agility

- Agility is **ability to move quickly and easily**.
- It is a property consisting of **quickness, lightness, & ease of movement**.
- The ability to **create and respond to change** in order to profit in a **turbulent(disordered) global business environment**.
- The ability to **quickly reprioritize use of resources** when requirements, technology, and knowledge shift.
- A very **fast response to sudden market changes** and emerging threats by intensive customer interaction.

Agility


- Use of evolutionary, incremental, and iterative delivery to converge on an optimal customer solution.
- Maximizing BUSINESS VALUE with right sized, just-enough, and just-in-time processes and documentation.

Current Functionality



Change Request

Effective response to change



Organizing a team so that it is in control to perform the work

Effective communication among all stakeholders

What is Agility?

Software Development Team



Drawing the
customer onto
the team

Eliminate the
“us and them”
attitude

Rapid and Incremental delivery of software

Not like this....



1



2



3



4

Like this!



1



2



3



4



5

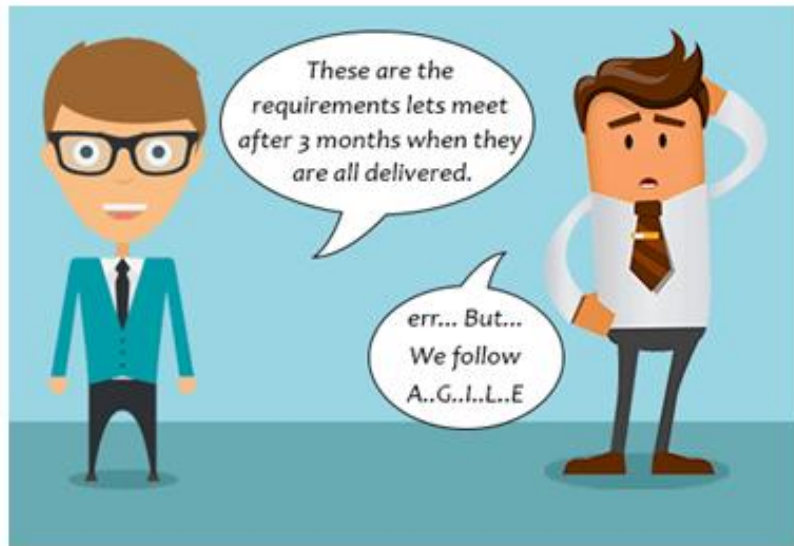
Agile Process

- Agile software process addresses **few assumptions**.
- **Difficulty in predicting changes** of requirements and customer priorities.
- For many types of software; **design and construction** are **interleaved** (mixed).
- **Analysis, design, construction** and **testing** are **not** as **predictable** as we might like.
- An agile **process** must be **adaptable**
- Requires **customer feedback**.

Agility Principles

- **Highest priority** is to **satisfy** the **customer** through **early & continuous delivery** of software
- **Welcome changing** requirements
- **Deliver** working software **frequently**
- **Business people** and **developers** must **work together**
- **Build** projects **around motivated** individuals
- Emphasize **face-to-face conversation**
- **Working software** is the **measure of progress**
- Continuous **attention** to **technical excellence** and **good design**
- **Simplicity** – the art of maximizing the amount of work done
- The best designs emerge from **self-organizing teams**
- The **team tunes** and **adjusts** its **behavior** to become more effective

Agility not work



Project plan & requirements are clear & unlikely to change



Unclear understanding of Agile Approach among Teams

Agility not work



Big Enterprises where team
collaboration is tough

Agile Process Models

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Feature Driven Development (FDD)
- Crystal
- Agile Modeling (AM)

Extreme Programming (XP)

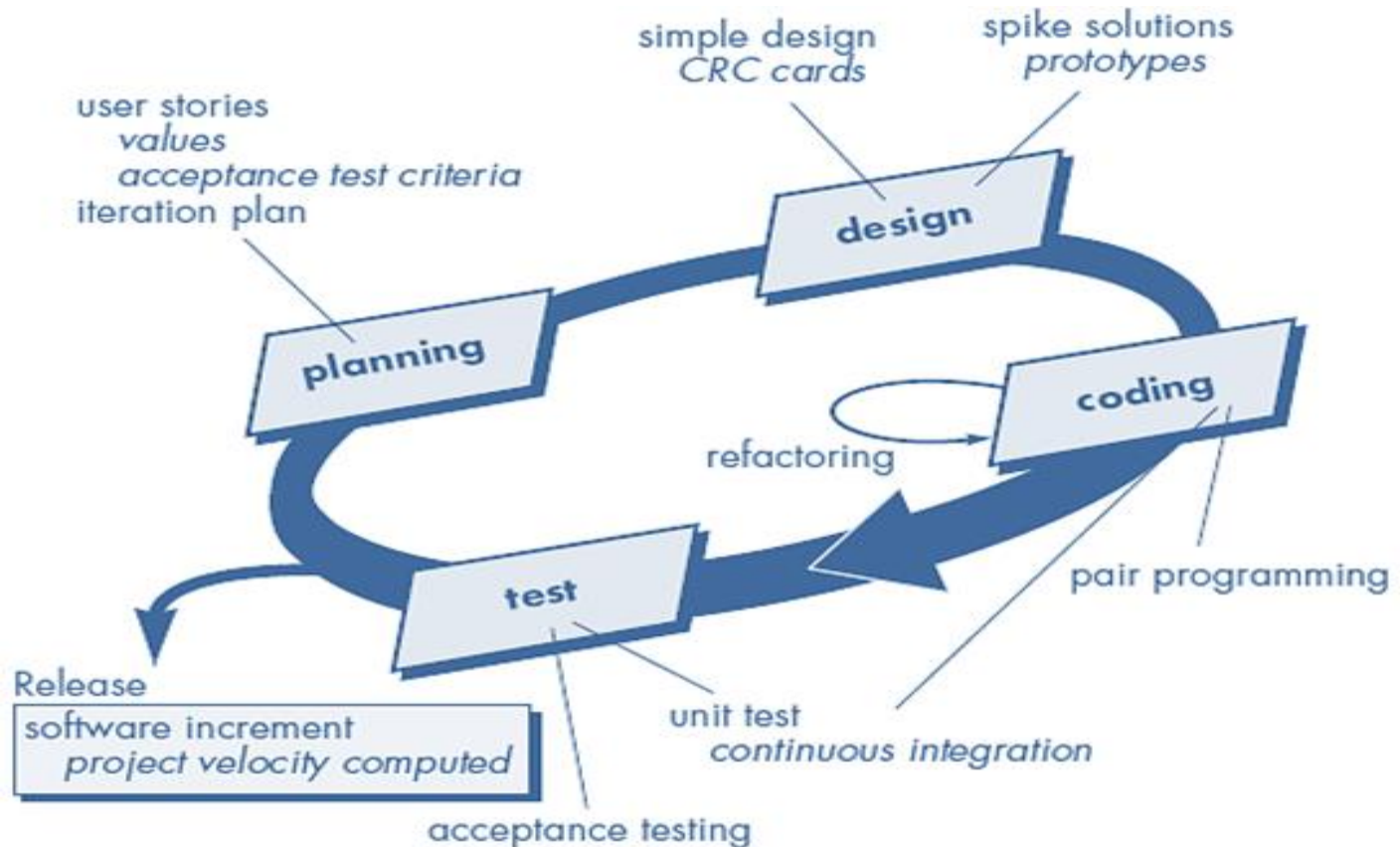
- The most widely used approach to agile software development
- A variant of XP called **Industrial XP (IXP)** has been proposed to target process for large organizations
- It uses **object oriented approach** as its preferred development model

XP Values

- **Communication:** To achieve **emphasized close & informal** customers and developers.
- **Simplicity:** It restricts developers to **design for immediate needs not for future needs**
- **Feedback:** It is derived **from** three sources the **implemented software**, the **customer** and **other software team members**, it uses **Unit testing** as primary testing
- **Courage:** It demands courage (discipline), there is often significant pressure to design for future requirements, XP team **must have the discipline (courage) to design for today.**
- **Respect:** XP team **respect** among **members**

The XP Process

- It considers four framework activities
- 1. Planning 2. Design 3. Coding 4. Testing



Planning



- User Stories
 - **Customers assigns value** (priority)
 - **Developers assigns cost** (number of development weeks)
- Project velocity
 - Computed at the end of first release
 - **Number of stories implemented in first release**
 - Estimates for future release
 - **Guard against over-commitment**

Design

CRC card

Class Name	
Responsibilities	Collaborators

- **Keep-it-Simple** (Design of extra functionality is discouraged)
- **Preparation of CRC card** is work project
 - CRC cards identify and organize object oriented classes
- **Spike Solutions**
 - Operational prototype intended to clear confusion
- Refactoring
- Modify internals of code, No observable change

Coding



- **Develops** a series of **Unit test** for stories included in current release
- **Complete code** perform **unit-test** to get immediate feedback
- XP recommend **pair-programming**, “**Two heads are better than one**”
- **Integrate code** with other team members, this “**continuous integration**” helps to avoid compatibility & interfacing problems, “**smoke testing**” environment to uncover errors early

Testing



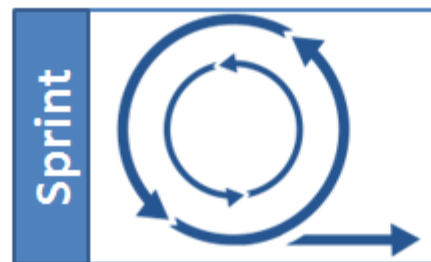
- **Unit test** by **developers** & fix small problems
- **Acceptance tests** - Specified by **customer**

What is scrum?

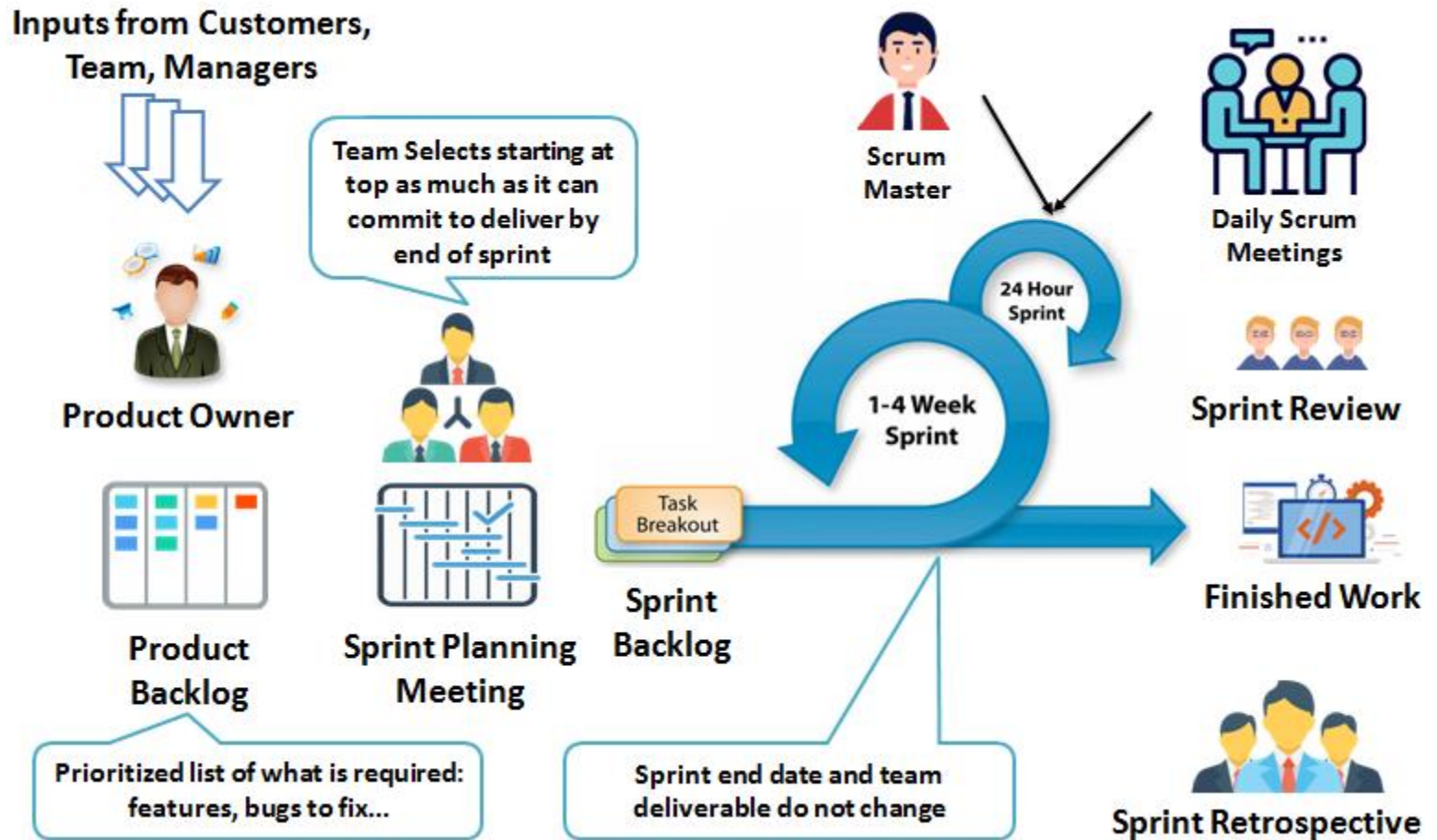


A scrum is a method of restarting play in rugby that involves players packing closely together with their heads down and attempting to gain possession of the ball. ...

- **Scrum** is an agile process model which is used for developing the complex software systems.
- It is a lightweight process framework.
- Lightweight means the overhead of the process is kept as small as possible in order to maximize the productivity.



Framework of scrum



Framework of scrum

- **1. Backlog**

- It is a **prioritized list of project requirements** or features that must be provided to the customer.
- The **items can be included** in the backlog at **any time**.
- The **product manager analyses** this **list** and **updates** the **priorities** as per the requirements.

- **2. Sprint**

- These are the **work units** that are needed **to achieve** the requirements mentioned in the backlogs.
- Typically the sprints have **fixed duration** or time box (of **2 to 4 weeks, 30 days**).
- **Change** are **not introduced** during the **sprint**.
- Thus sprints allow the team **members** to **work in stable** and **short-term environment**.

Framework of scrum

- **3. Scrum Meetings**
- There are **15 minutes daily meetings** to **report the completed activities, obstacles** and **plan for next activities**.
- Following are three questions that are mainly discussed during the meetings.
- **1. What are the tasks done since last meeting ?**
- **2. What are the issues that team is facing ?**
- **3. What are the next activities that are planned?**
- The **scrum master** leads the meeting and **analyses the response** of each team member.
- Scrum meeting **helps the team to uncover potential problems** as early as possible
- It leads to **“knowledge socialization”** & promotes **“self-organizing team structure”**

Framework of scrum

- **4. Demo**
- Deliver **software increment** to customer.
- Implemented functionalities are **demonstrated** to the customer.

Adaptive Software development (ASD)

- This is a technique for building complex software systems using iterative approach.
- ASD focus on working in collaboration and team self-organization.
- ASD incorporates three phases:
- 1. Speculation 2. Collaboration 3. Learning

Adaptive Software development (ASD)

Speculation (ASD)

- The adaptive **cycle planning** is **conducted**.
- In this cycle planning mainly three types of information is used
- Customer's **mission statement**
- Project **constraints** like Delivery date, budgets etc...
- **Basic requirements** of the project

Adaptive Software development (ASD)

Collaboration (ASD)

- In this, **collaboration** among the **members** of **development team** is a key factor.
- For **successful collaboration** and coordination it is necessary to have following **qualities** in every individual
- **Assist each other** without resentment (offense)
- **Work hard**
- **Posses** the required **skill set**
- **Communicate problems** and help each other
- **Criticize** without any **hate**

Adaptive Software development (ASD)

Learning (ASD)

- Emphasize is on **learning** new **skills** and techniques.
- There are three ways by which the team members learn

1. **Focus groups**

- The **feedback** from the **end-users** is obtained.

2. **Formal technical review**

- This review is conducted for better quality.

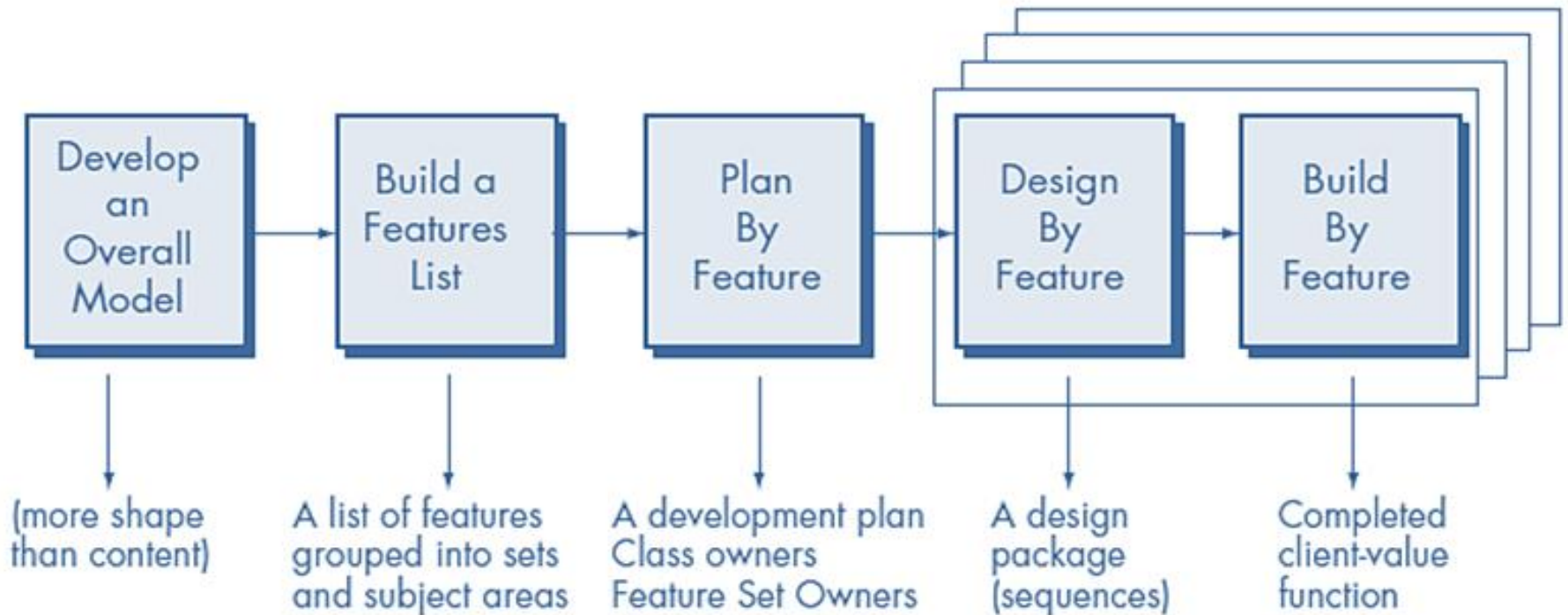
3. **Postmortems**

- Team analyses its own performance and makes appropriate improvements.

DSDM(Dynamic Systems Development Methods)

- Various phases of this life cycle model.
- **Feasibility study**
- By analyzing the business requirements and constraints the viability of the application is determined
- **Business study**
- The functional and informational requirements are identified and then the business value of the application is determined.
- **Functional model iteration**
- The incremental approach is adopted for development
- **Design and build iteration**
- If possible design and build activities can be carried out in parallel
- **Implementation**
- The software increment is placed in the working environment.

Feature Driven Development (FDD)



Feature Driven Development (FDD)

- It is practical process model for object oriented software engineering.
- In FDD, the feature means client valued function.
- Various phases in the FDD life cycle
- **1. Develop overall model**
- The high-level walkthrough of scope and detailed domain walkthrough are conducted to create overall models.
- **2. Build feature list**
- List of features is created and expressed in the following form
- <action> the <result> <by for of to> a(n) <object>
- For Ex. “Display product-specifications of the product”

Feature Driven Development (FDD)

- **3. Plan by feature**
- After completing the feature list the development plan is created
- **4. Design by feature**
- For each feature the sequence diagram is created
- **5. Build by feature**
- Finally the class owner develop the actual code for their classes

Crystal

- The Crystal methodology is one of the **most lightweight, adaptable approaches** to software development.
- Crystal is actually comprised of a family of agile methodologies such as Crystal Clear, Crystal Yellow, Crystal Orange and others, whose unique characteristics are driven by several factors such as **team size, system criticality, and project priorities**.
- This Crystal family addresses the realization that each project may require a slightly tailored set of policies, practices, and processes in order to meet the project's unique characteristics.
- Several of the key tenets of Crystal include **teamwork, communication, and simplicity, as well as reflection to frequently adjust and improve the process**.
- Like other agile process methodologies, Crystal promotes early, frequent delivery of working software, high user involvement, adaptability, and the removal of bureaucracy or distractions.

Agile modeling

- Agile Modeling (AM) is a practice-based methodology for effective modeling and documentation of software-based systems.
- Simply put, Agile Modeling (AM) is a collection of values, principles, and practices for modeling software that can be applied on a software development project in an effective and light-weight manner.
- Although AM suggests a wide array of “core” and “supplementary” modeling principles, those that make AM unique are:
- **Use multiple models.** There are many different models and notations that can be used to describe software. AM suggests that to provide needed insight, each model should present a different aspect of the system and only those models that provide value to their intended audience should be used.
- **Travel light.** As software engineering work proceeds, keep only those models that will provide long-term value and jettison the rest.
- **Content is more important than representation.** Modeling should impart information to its intended audience. A syntactically perfect model that imparts little useful content is not as valuable as a model with flawed notation that nevertheless provides valuable content for its audience.
- **Know the models and the tools you use to create them.** Understand the strengths and weaknesses of each model and the tools that are used to create it.
- **Adapt locally.** The modeling approach should be adapted to the needs of the agile team.

Assignment

Sr. No	Questions
1	Explain Agile development in detail or what are the key factors that are considered for agile development? (Asked in GTU)
2	List different agile process model. Explain any one with suitable example.(asked in GTU)
3	Myths of planned development.
4	Explain Extreme programming process.
5	What is scrum ? Explain in detail.(asked in GTU)
6	What are agile practices ? Explain each in detail.
7	Discuss agile modeling significance. What are different modeling principles that makes agile modeling unique.
8	Define agility. What are the things that an agile software should have?
9	Explain Adaptive software development model in detail.(asked in GTU)

References

- <https://study.com/academy/lesson/what-is-agile-software-development-definition-methodology.html>
- <https://www.youtube.com/watch?v=Z9QbYZh1YXY>
- <https://www.youtube.com/watch?v=apOvF9NVguA>
- <https://www.youtube.com/watch?v=EDT0HMtDwYI>
- <https://www.youtube.com/watch?v=v9tjYJXcdTs>