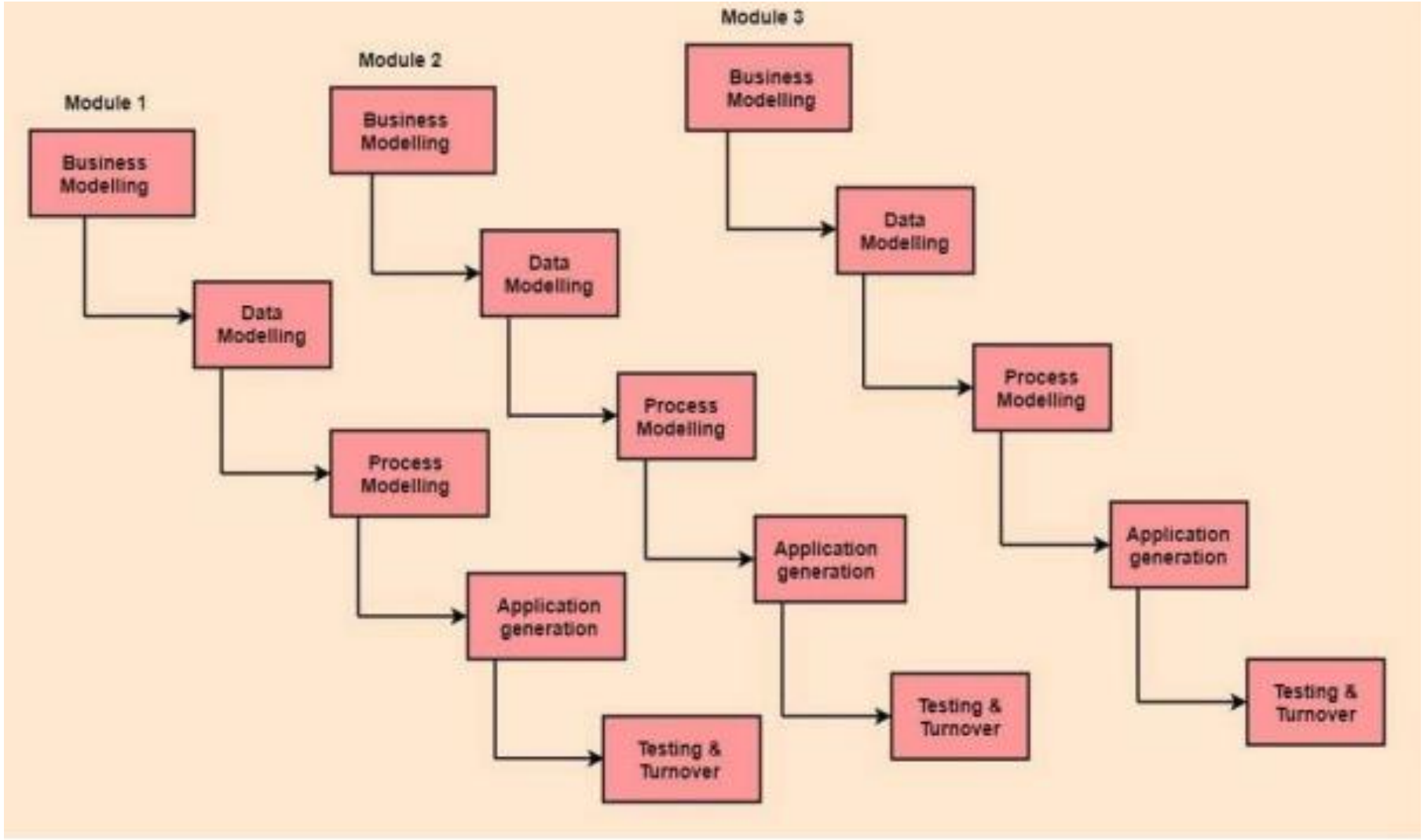
THE RAD (Rapid Application Development) MODEL

- An incremental software process model
- Having a short development cycle
- High-speed adoption of the waterfall model using a component based construction approach
- Creates a fully functional system within a very short span time of 60 to 90 days

THE RAD MODEL

- The RAD Model consists of the following phases:
- Communication Planning Construction
- Component reuses automatic code generation testing
- Modeling
- Business modeling Data modeling Process modeling
- Deployment integration delivery feedback

THE RAD MODEL



THE RAD MODEL

- Multiple software teams work in parallel on different functions
- Modeling encompasses three major phases: Business modeling, Data modeling and process

modeling

- Construction uses reusable components, automatic code generation and testing

Problems in RAD

- Requires a number of RAD teams
- Requires commitment from both developer and customer for rapid-fire completion of activities
- Requires modularity
- Not suited when technical risks are high

THE CONCURRENT DEVELOPMENT MODEL

- Also called concurrent engineering
- Constitutes a series of framework activities, software engineering action, tasks and their associated states
- All activities exist concurrently but reside in different states
- Applicable to all types of software development
- Event generated at one point in the process trigger transitions among the states

A FINAL COMMENT ON EVOLUTIONARY PROCESS

- Difficult in project planning
- Speed of evolution is not known

Does not focus on flexibility and extensibility (more emphasis on high quality)

- Requirement is balance between high quality and flexibility and extensibility

CMMI
Capability Maturity Model Integration

What is CMMI?

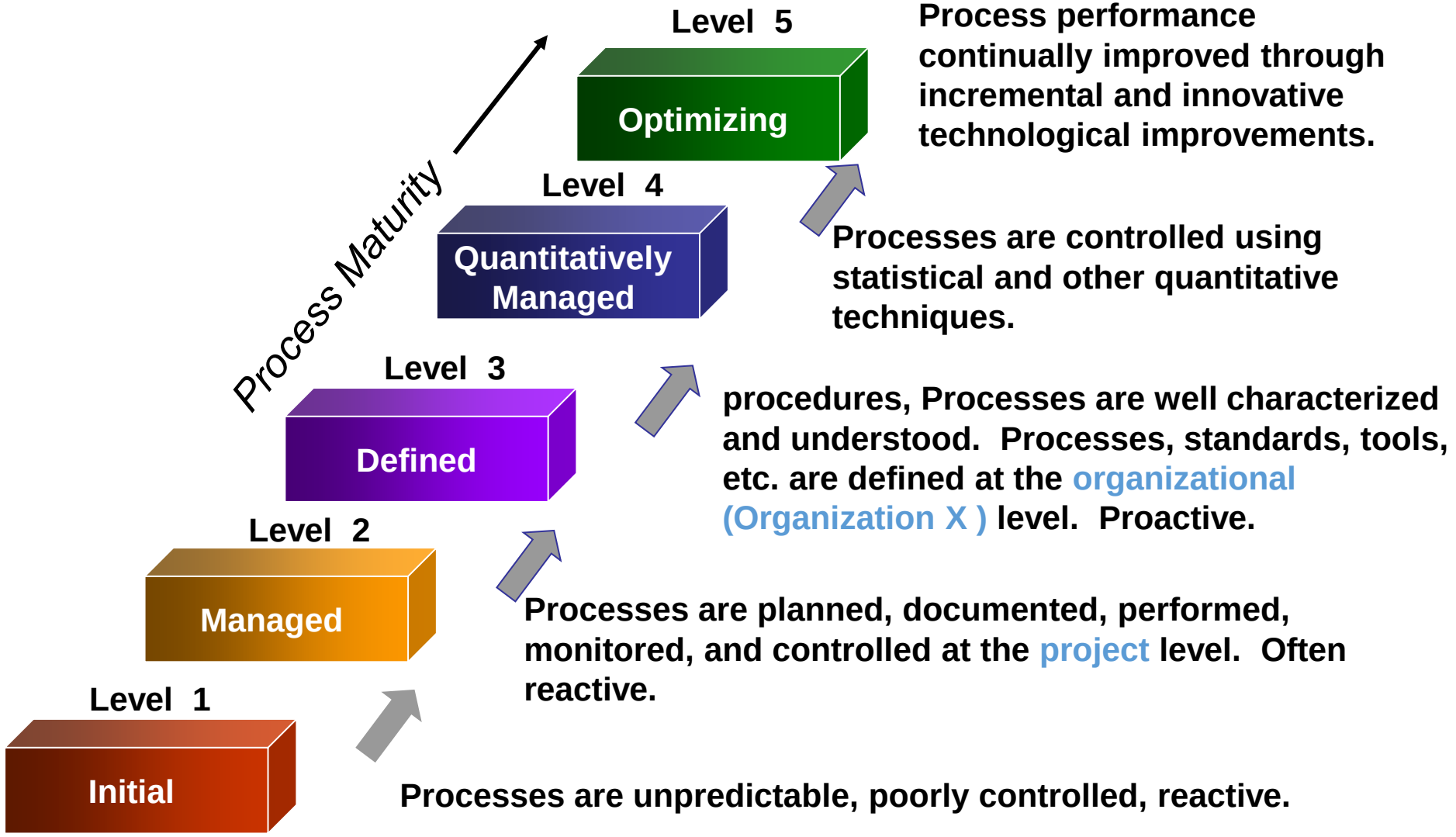
- CMMI (Capability Maturity Model Integration) is a proven industry framework to improve product quality and development efficiency for both hardware and software
  - Many companies have been involved in CMMI definition such as Motorola and Ericsson
  - CMMI has been established as a model to improve business results
- CMMI, staged, uses 5 levels to describe the maturity of the organization, same as predecessor CMM
  - Vastly improved version of the CMM
  - Emphasis on business needs, integration and institutionalization

CMMI Models within the Framework

- Models:
  - **Systems Engineering + Software Engineering (SE/SW)**
  - Systems Engineering + Software Engineering + Integrated Product and Process Development (IPPD)
  - Systems Engineering + Software Engineering + Integrated Product and Process Development + Supplier Sourcing (SS)
  - Software Engineering only
- Representation options:
  - **Staged**
  - Continuous
- The CMMI definition of "Systems Engineering" -

  "The interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations and constraints into a product solution and to support that solution throughout the product's life." **This includes both hardware and software.**

CMMI Staged Representation - 5 Maturity Levels



Maturity Level 1
Initial

- Maturity Level 1 deals with performed processes.
- Processes are unpredictable, poorly controlled, reactive.
- The process performance may not be stable and may not meet specific objectives such as quality, cost, and schedule, but useful work can be done.

Maturity Level 2
Managed at the Project Level

- Maturity Level 2 deals with managed processes.
- A managed process is a performed process that is also:
  - Planned and executed in accordance with policy
  - Employs skilled people
  - Adequate resources are available
  - Controlled outputs are produced
  - Stakeholders are involved
  - The process is reviewed and evaluated for adherence to requirements
- Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.
- The managed process comes closer to achieving the specific objectives such as quality, cost, and schedule.

Maturity Level 3
Defined at the Organization Level

- Maturity Level 3 deals with defined processes.
- A defined process is a managed process that:
  - Well defined, understood, deployed and executed across the entire **organization**. Proactive.
  - Processes, standards, procedures, tools, etc. are defined at the organizational level. Project or local tailoring is allowed, however it must be based on the organization's set of standard processes and defined per the organization's tailoring guidelines.
- Major portions of the organization cannot "opt out."

Behaviors at the Five Levels

Maturity Level	Process Characteristics	Behaviors
5 Optimizing	Focus is on continuous quantitative improvement	Focus on "fire prevention"; improvement anticipated and desired, and impacts assessed.
4 Quantitatively Managed	Process is measured and controlled	Greater sense of teamwork and inter-dependencies
3 Defined	Process is characterized for the organization and is proactive	Reliance on defined process. People understand, support and follow the process.
2 Managed	Process is characterized for projects and is often reactive	Over reliance on experience of good people – when they go, the process goes.
1 Initial	Process is unpredictable, poorly controlled, and reactive	Focus on "fire fighting"; effectiveness low – frustration high.

CMMI Resources

- Software Engineering Institute's CMMI website:
  http://www.sei.cmu.edu/cmmi/

Agile Model

- Mostly used model in todays digital era.
- Agile means” The ability to respond to(continue) changes from requirements, technology and people”
- It is an incremental and iterative process of software development.

When to use the Agile Model?

- When Project size is large
- When frequent changes are required.
- When a highly qualified and experienced team is available.
- When a customer is ready to have a meeting with a software team all the time.(every iteration)
- Projects with flexible timeline and budget.

Agile Principles

1. Highest priority is to satisfy the customers to early and continue delivery of software.
2. Being flexible about changing requirement at any point of development.
3. Working on frequent and short deliveries like couple of weeks or months with preference.
4. Transparency between business people and developers and requires them to work together.(all stakeholder work together)
5. By providing a better productive environment and providing them with all the support and motivation it leads to better productivity.
6. Face to face communication as the most effective way to communicate between customer and development team(feedback)
7. Continuous attention towards effective designing and technical excellence through optimal code standard.
8. It promotes sustainable development (rapid delivery)by developers, users and sponsors should work together.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity is the art of maximum result and less hard work by removing unnecessary tasks and prioritizing activities.(focus on design and coding, less documentation)
11. The best architectures , requirements and designs emerge from self- organizing and experience teams.
12. For development effective software, regular analysis and work on improving the overall delivery or the development process.

Advantages/Disadvantages of Agile model

Advantages of Agile model

- Support customer involvement and customer satisfaction
- Strong communication of the software team with the customer’
- Little planning require(more focus on design and development)
- Efficient design and fulfils the business requirement.
- Anytime changes are acceptable.
- Provides a very realistic(accurate) approach to software development
- Updated version of functioning software are released every week.
- it reduces total development time.

Disadvantages of Agile model

- Due to the lack of proper documentation (cant waste time on doc), once the project completes and the developers allotted to another project , maintenance of the finished project can become difficult.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

Introduction To Agile Tool: Jira

- - JIRA is a tool developed by Australian Company Atlassian.
- - It is used for bug tracking, issue tracking, and project management.
- - The name "JIRA" is actually inherited from the Japanese word "Gojira" which means "Godzilla".
- - The basic use of this tool is to track issue and bugs related to your software and Mobile apps.
- - JIRA is an incident managementtool.
- - JIRA is a platform independent tool; it can be used with any OS.
- - JIRA is multi-lingual tool – English,French,German,Japanese,Spanish,etc.
- - JIRA supports MySQL,Oracle, PostgreSQL and SQL server in the backend.
- - JIRA can be integrated with many other tools – Subversion,GIT,Clear case, and many more

Extreme Programming:

- Extreme Programming (XP), the most widely used approach to agile software development. XP proposed by kent beck during the late 1980’s.

XP Values

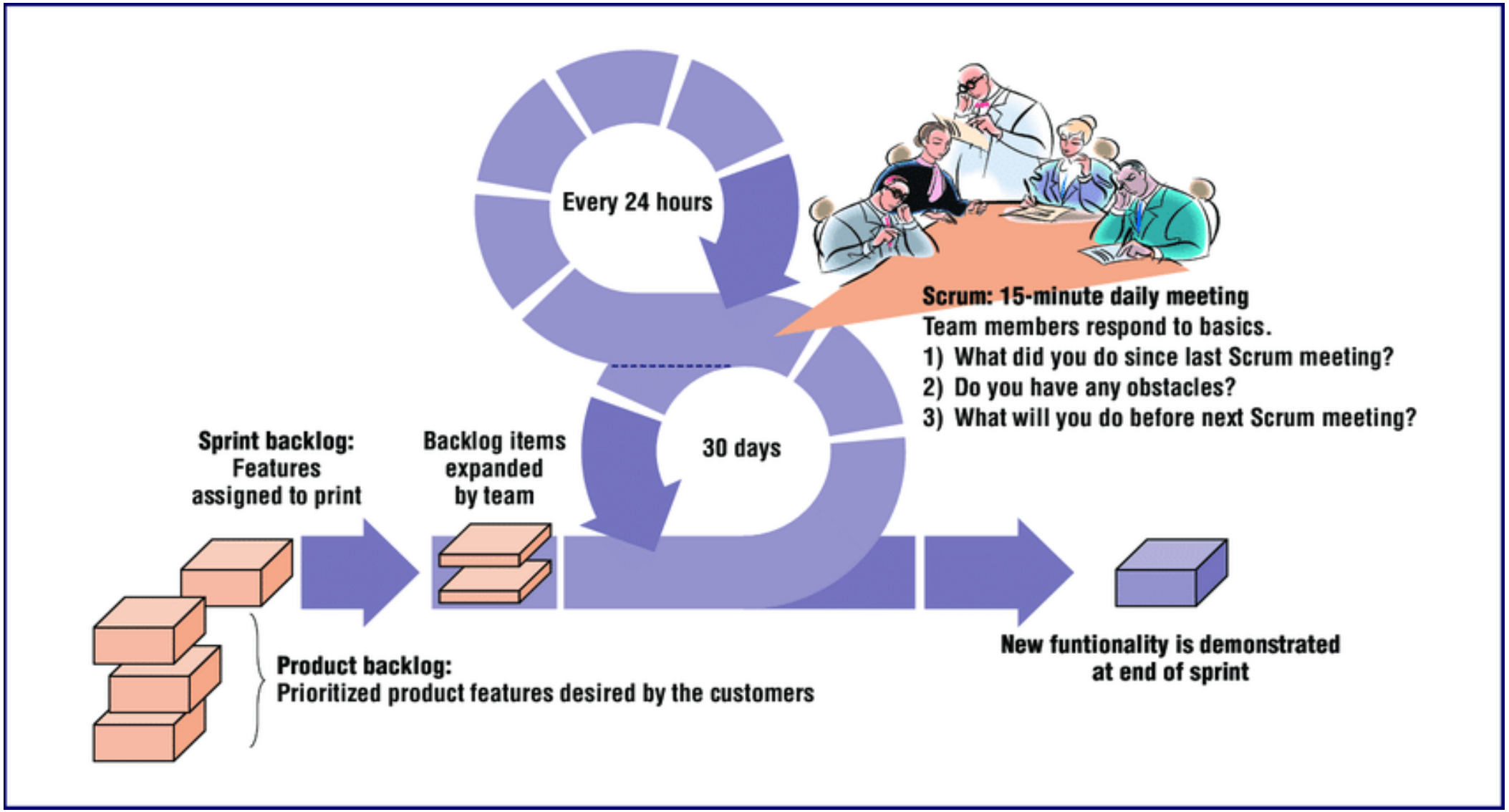
- Beck defines a set of five values —communication, simplicity, feedback, courage,and respect. Each of these values is used in XP activities, actions, and tasks.

  - Effective communication between software engineers and other stakeholders .
  - . To achieve simplicity, XP restricts developers to design only for immediate needs, rather than future needs.
  - Feedback is derived from three sources: the implemented software itself, the customer, and other software team members
  - courage.(discipline) An agile XP team must have the discipline (courage) to design for today, recognizing that future requirements may change dramatically.
  - the agile team inculcates respect among it members, between other stakeholders and team members.

Scrum

- Scrum is an agile software development method that was coined by Jeff Sutherland and his development team in the early 1990's.
- Scrum has the following framework activities: requirements, analysis,design, evolution, and delivery.
- Within each framework activity, actions and work tasks occur within a process called a sprint.
- scrum defines a set of development actions:
- Backlog—a prioritized list of project requirements or features that provide business value for the customer. Items can be added to the backlog at any time (this is how changes are introduced). The product manager assesses the backlog and updates priorities as required.
- Sprints—consist of work units that are required to achieve a requirement defined in the backlog .

Scrum process flow



- Scrum meetings—are short (typically 15 minutes) meetings held daily by the

Scrum team. Three key questions are asked and answered by all team members.

  - What did you do since the last team meeting?
  - What obstacles are you encountering?
  - What do you plan to accomplish by the next team meeting?

- A team leader, called a Scrum master, leads the meeting and assesses the responses from each person.
- Demos—deliver the software increment to the customer so that functionality that has been implemented can be demonstrated and evaluated by the customer.