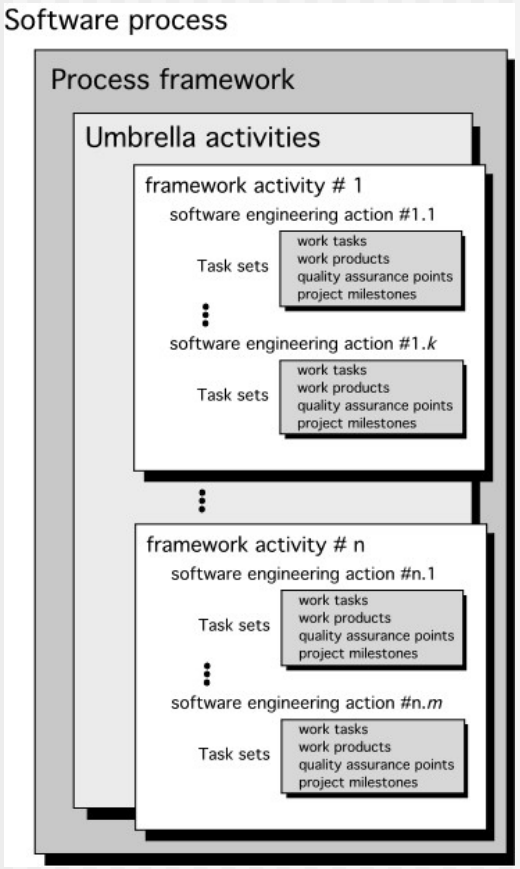
Lecture 2

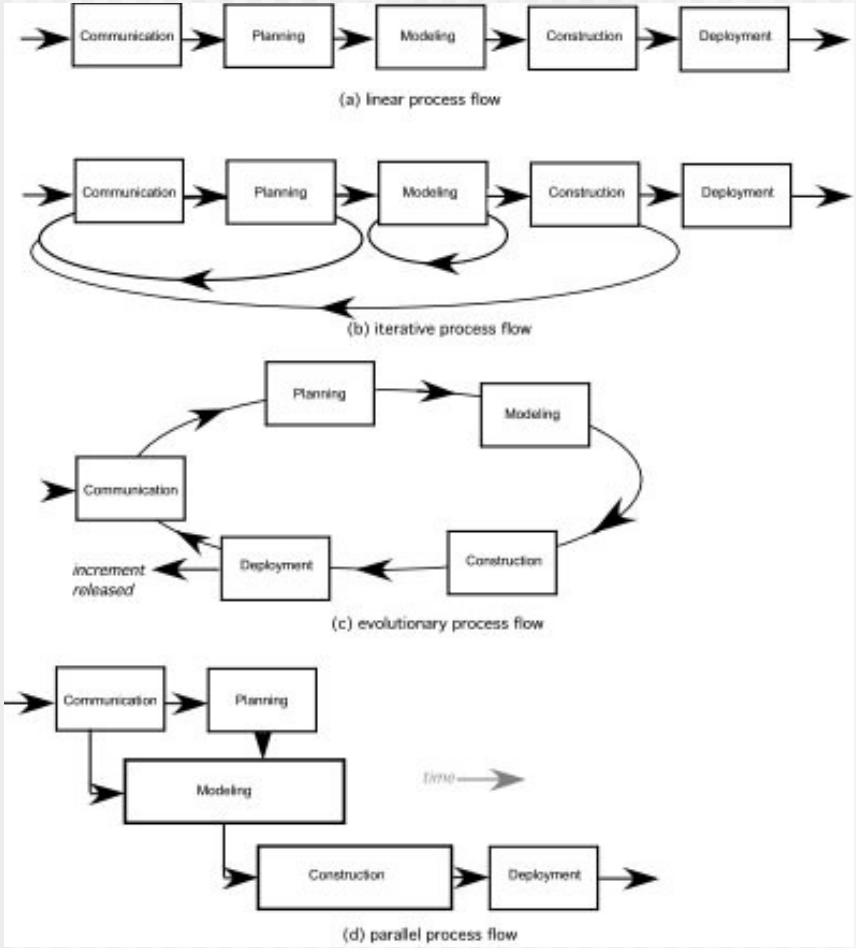
- **Software Process**

“What are the activities in Software Engineering?”

A Generic Process Model



Process Flow



Identifying a Task Set

- A task set defines the actual work to be done to accomplish the objectives of a software engineering action.
  - A list of the task to be accomplished
  - A list of the work products to be produced
  - A list of the quality assurance filters to be applied

Process Patterns

- A *process pattern*
  - describes a process-related problem that is encountered during software engineering work,
  - identifies the environment in which the problem has been encountered, and
  - suggests one or more proven solutions to the problem.
- Stated in more general terms, a process pattern provides you with a *template* [Amb98]—a consistent method for describing problem solutions within the context of the software process.

Process Pattern Types

- *Stage patterns*—defines a problem associated with a framework activity for the process.
- *Task patterns*—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice
- *Phase patterns*—define the sequence of framework activities that occur with the process, even when the overall flow of activities is iterative in nature.

Process Assessment and Improvement

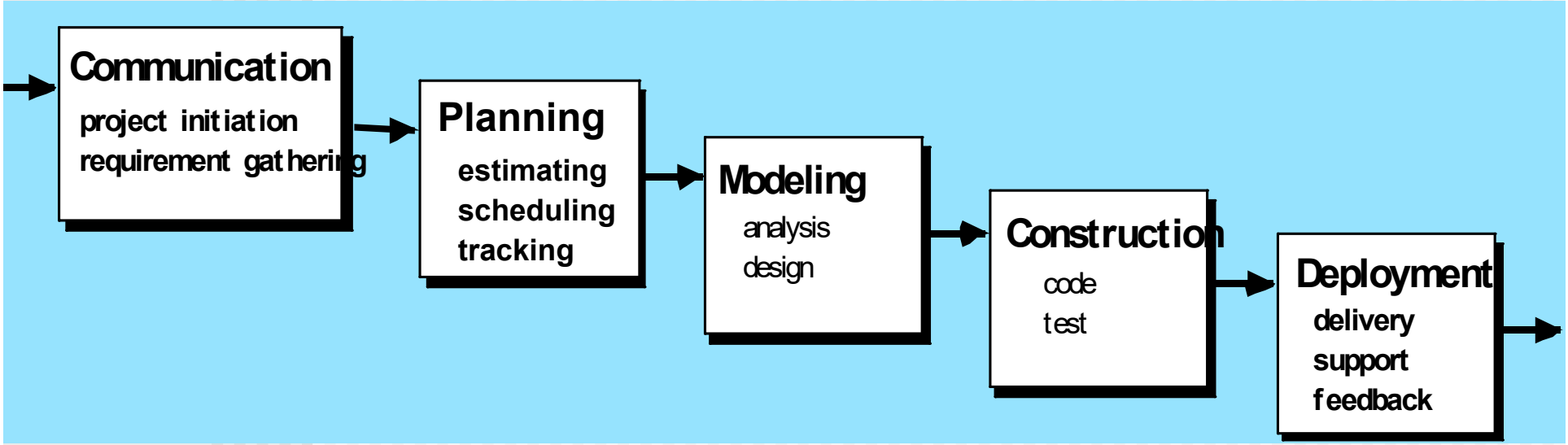
- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — provides a five step process assessment model that incorporates five phases: initiating, diagnosing, establishing, acting and learning.
- **CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**—provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment [Dun01]
- **SPICE—The SPICE (ISO/IEC15504)** standard defines a set of requirements for software process assessment. The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process. [ISO08]
- **ISO 9001:2000 for Software—**a generic standard that applies to any organization that wants to improve the overall quality of the products, systems, or services that it provides. Therefore, the standard is directly applicable to software organizations and companies. [Ant06]

Prescriptive Process Models

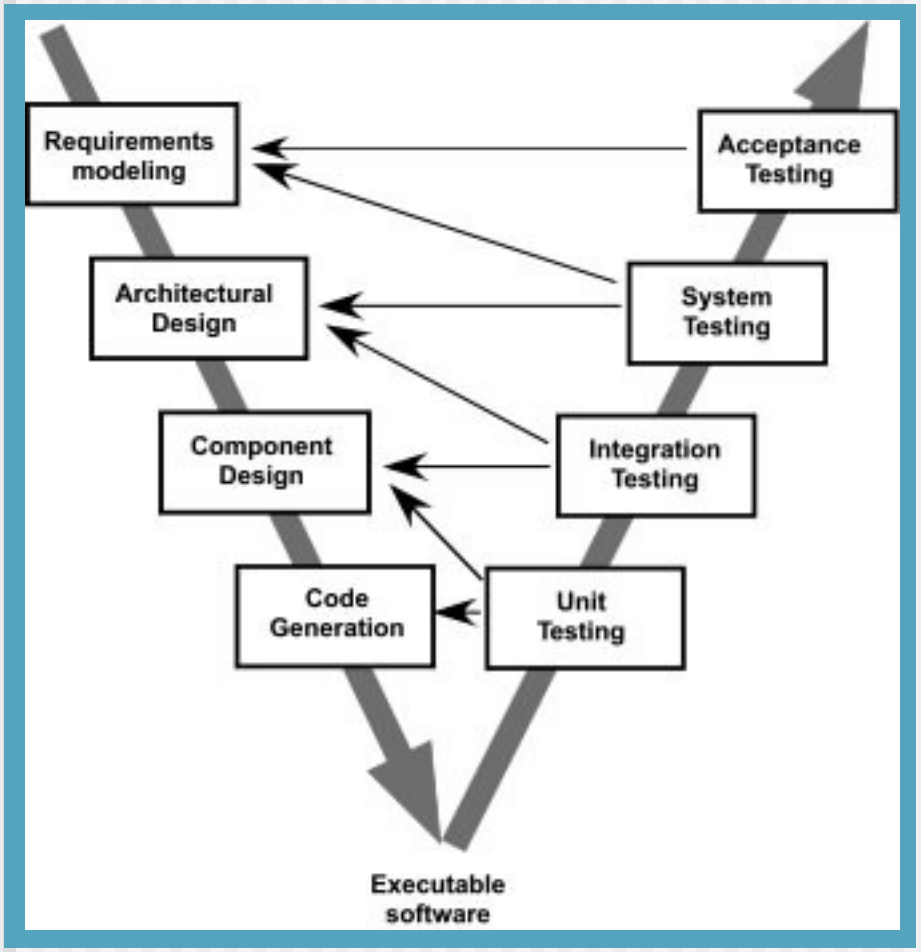
- Prescriptive process models advocate an orderly approach to software engineering:
  - **The Waterfall Model**
  - **The V-Model**
  - **The Incremental Model**
  - **The Evolutionary Model**

The Waterfall Model

- Classic, systematic and sequential life cycle.
- A phase has to be complete before moving onto the next phase
- Difficult to respond to the changes from customer requirements



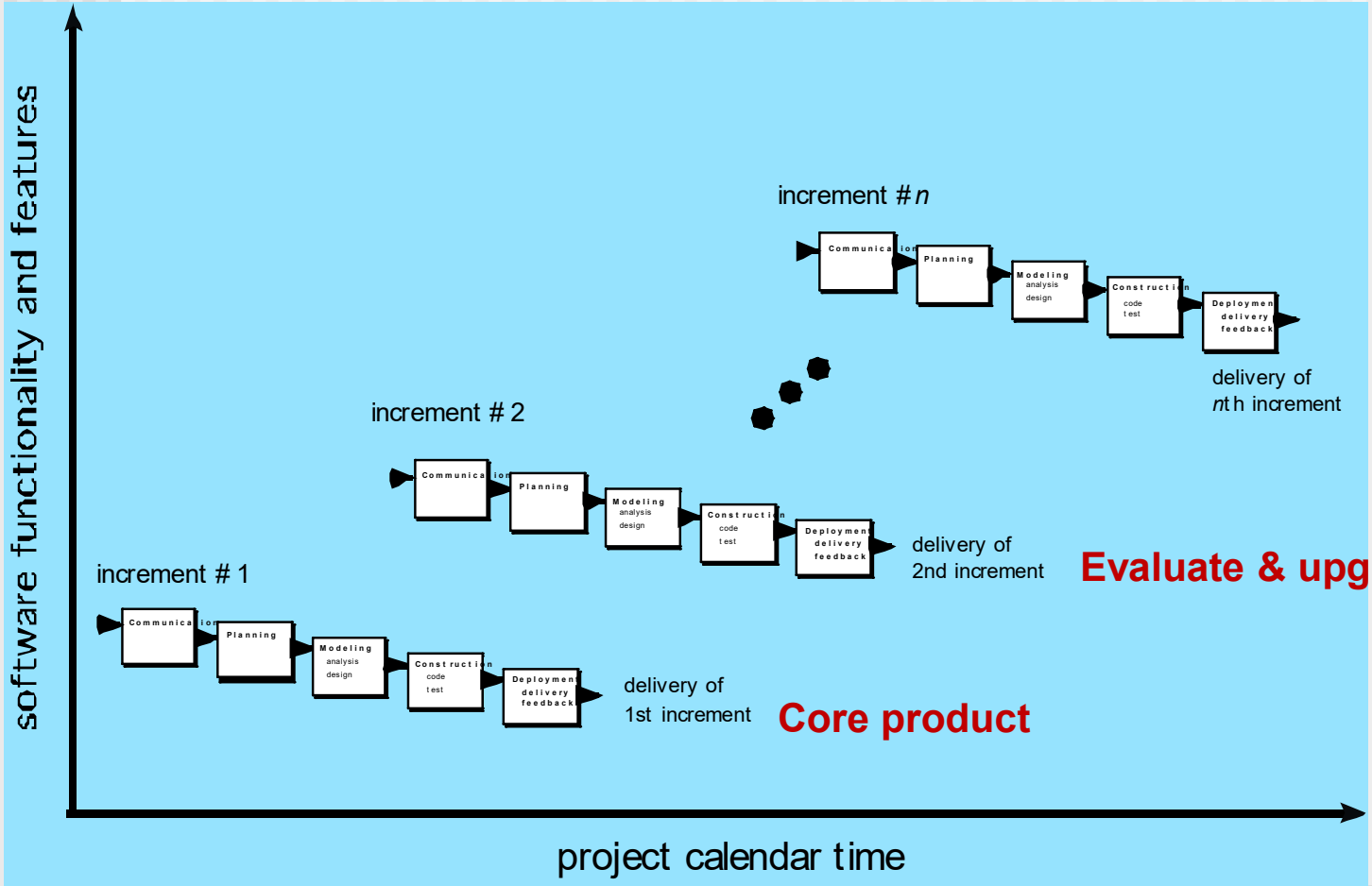
The V-Model



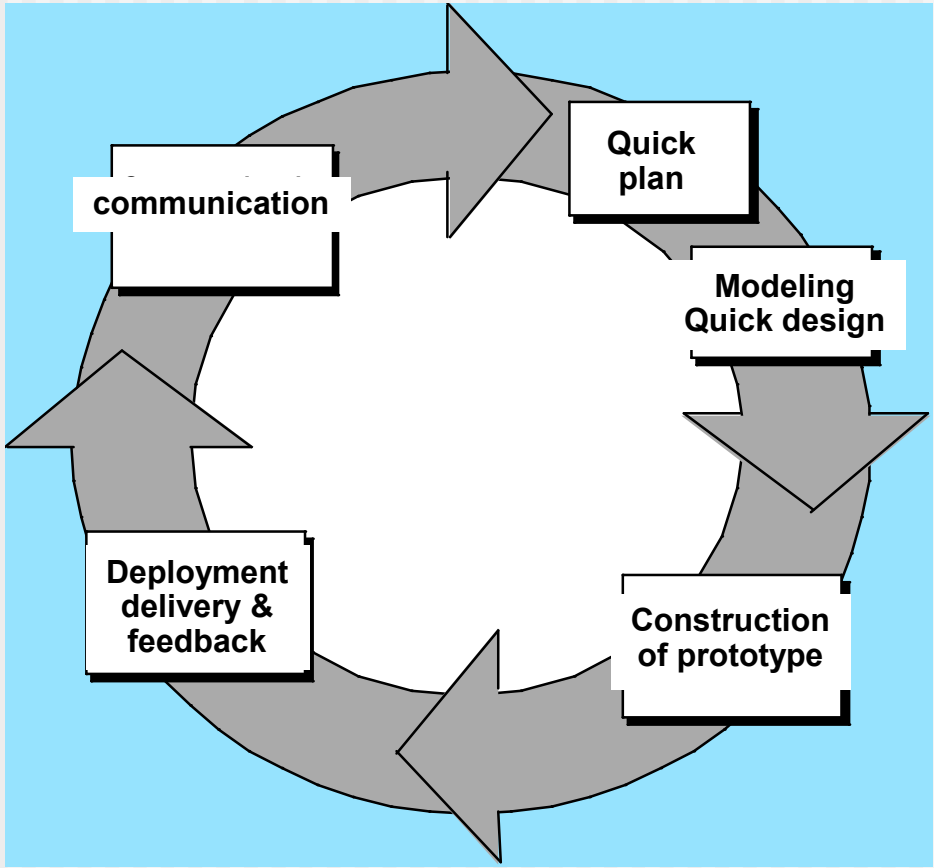
Quality Assurance actions

The Incremental Model

- Combines linear and parallel process flows.

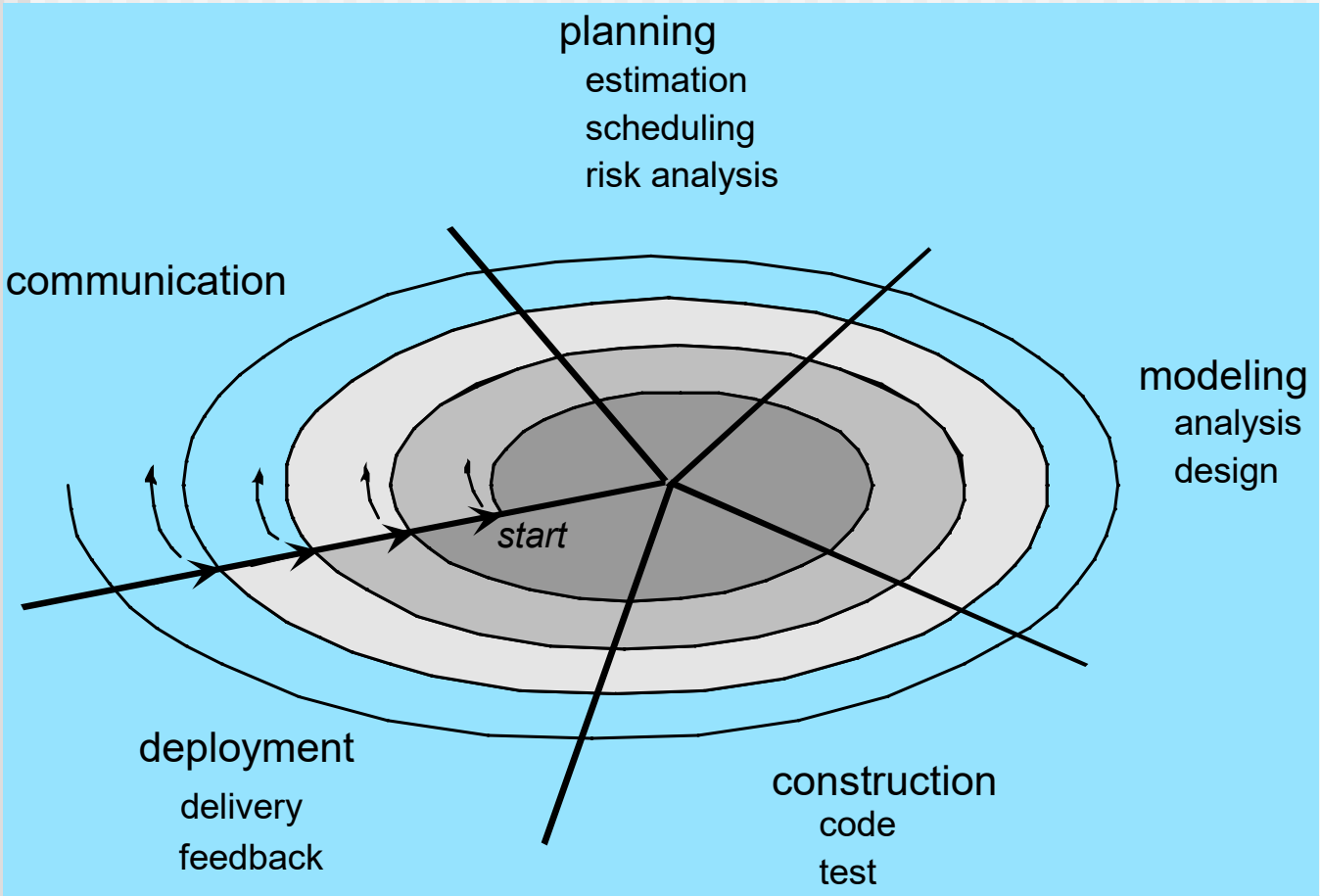


Evolutionary Models: 1. Prototyping

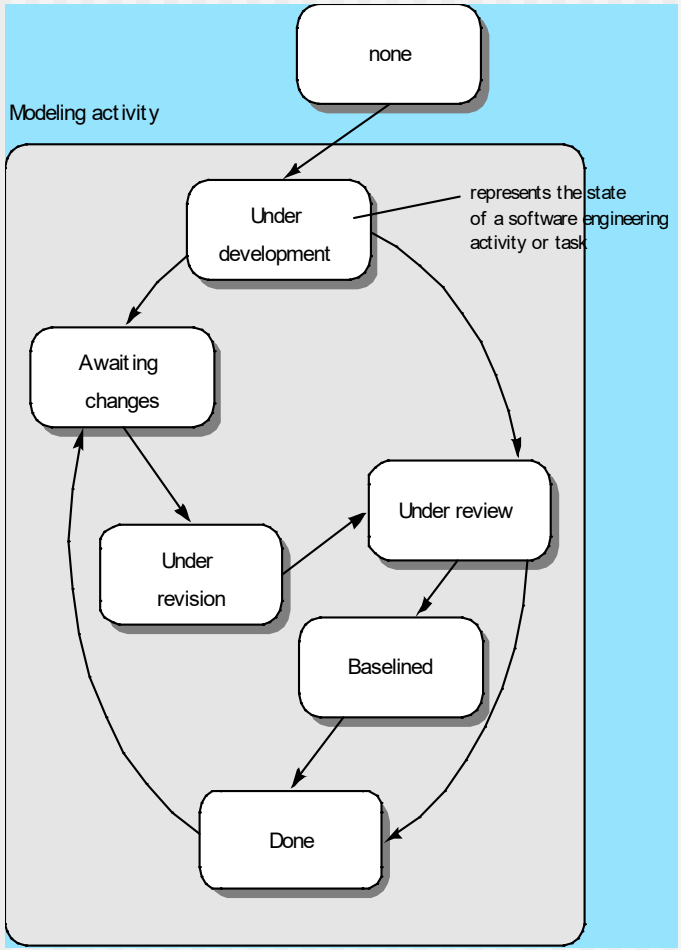


Evolutionary Models: 2. The Spiral

- Couples iterative nature of prototyping with controlled and systematic aspects of waterfall model.



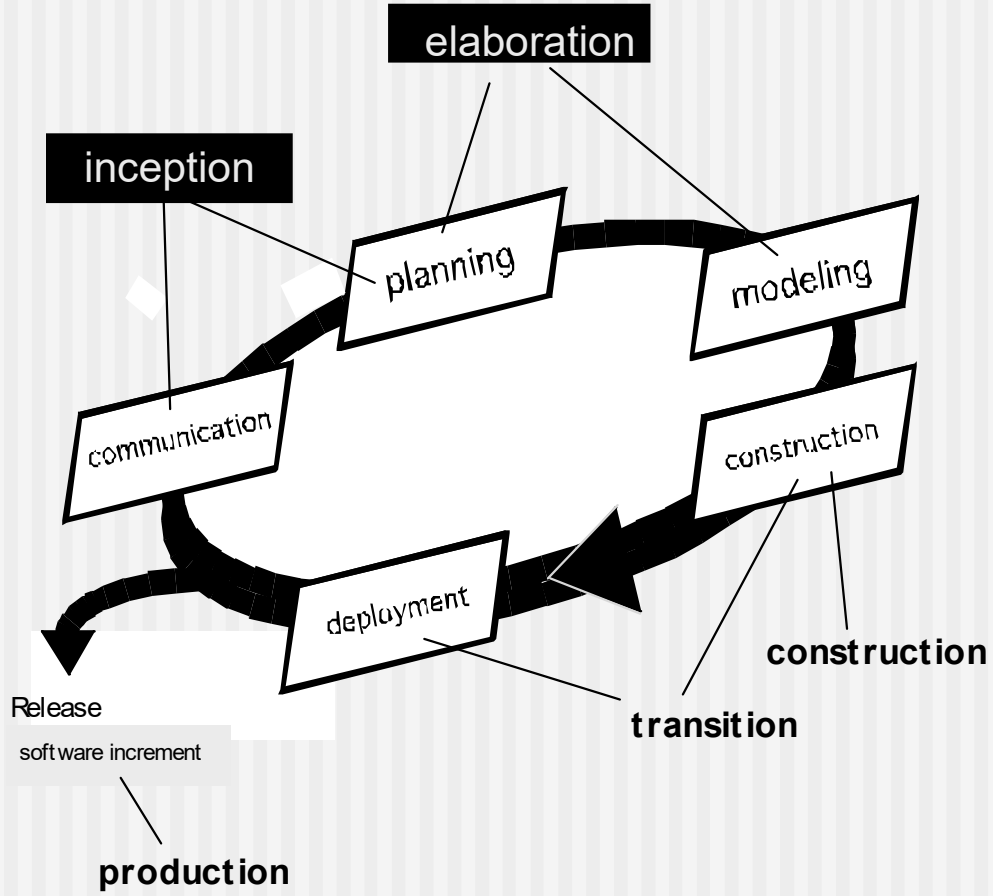
Evolutionary Models: 3. Concurrent



Specialized Process Models

- **Component-based Development**—The process to apply when reuse is a development objective.
- **Formal Methods**—Emphasizes the mathematical specification of requirements.
- **Aspect-Oriented Software Development**—Provides a process and methodological approach for defining, specifying, designing, and constructing *aspects.*
- **Unified Process**—A "use-case driven, architecture-centric, iterative and incremental" software process closely aligned with the Unified Modeling Language (UML).

The Unified Process (UP)



The Manifesto for Agile Software Development

"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**
- ***Customer collaboration* over contract negotiation**
- ***Responding to change* over following a plan**

That is, while there is value in the items on the right, we value the items on the left more."

Kent Beck et al

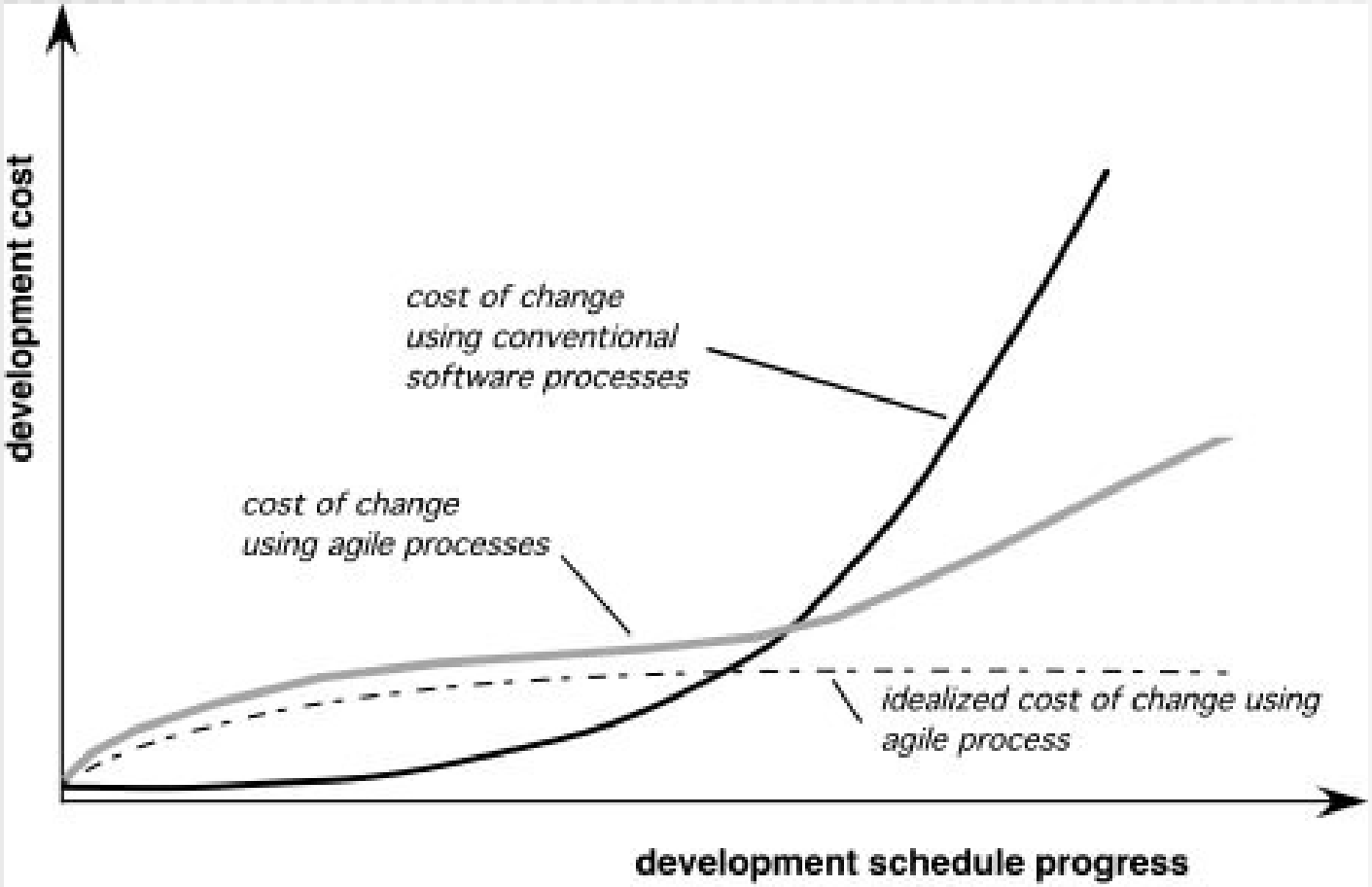
What is “Agility”?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding …

- Rapid, incremental delivery of software

Agility and the Cost of Change



An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple ‘software increments’
- Adapts as changes occur

Agility Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face–to–face conversation.

Agility Principles

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self–organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Types of Agile Process

- Extreme Programming (XP)
- Industrial XP (IXP)
- Scrum
- Dynamic Systems Development Method

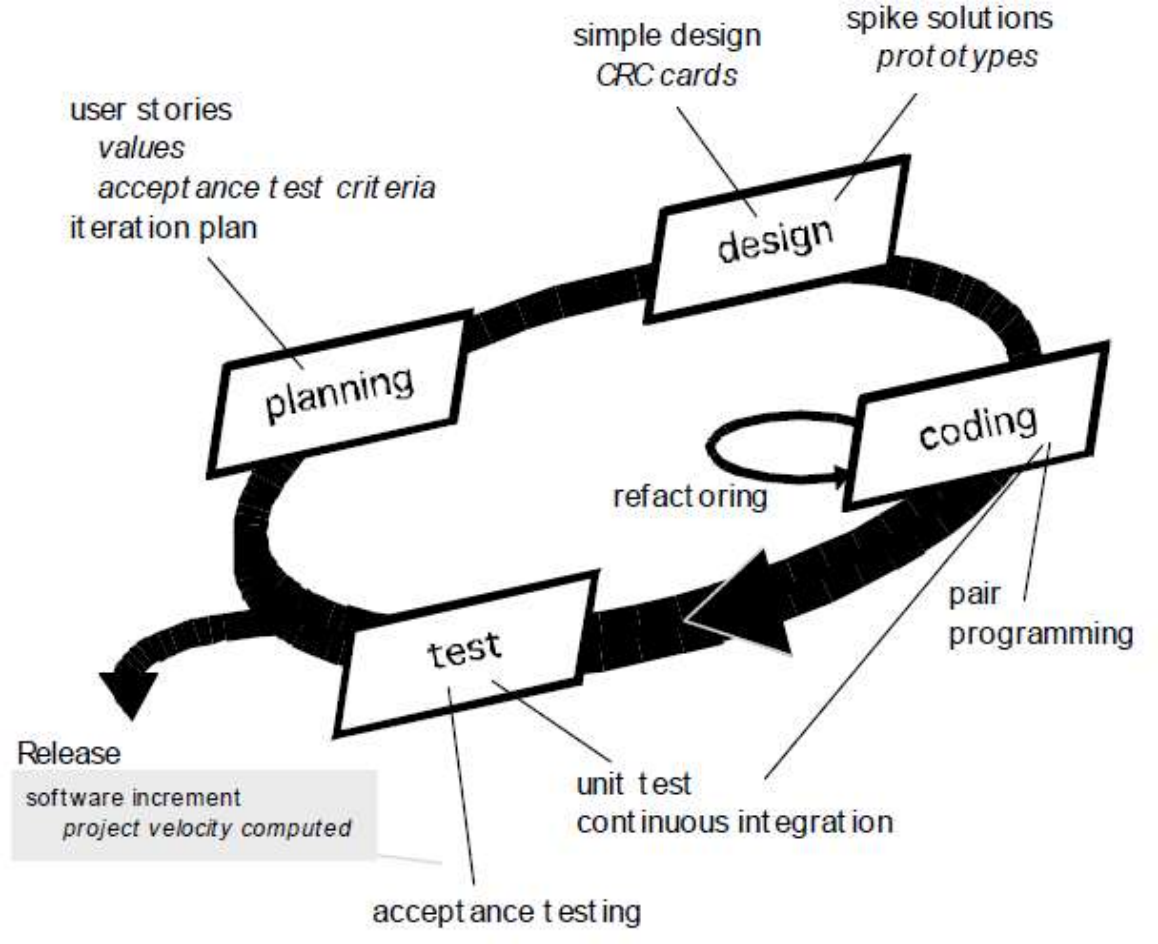
Extreme Programming (XP)

- The most widely used agile process, originally proposed by Kent Beck.
- **XP Planning**
  - Begins with the creation of "**user stories**".
  - Agile team assesses each story and assigns a **cost**.
  - Stories are grouped to for a **deliverable increment**.
  - A **commitment** is made on delivery date
  - After the first increment, "**project velocity**" is used to help define subsequent delivery dates for other increments.

Extreme Programming (XP)

- **XP Design**
  - Follows the **KIS (Keep It Simple) principle**.
  - Encourage the use of **CRC (class-responsibility-collaborator) cards**.
  - For difficult design problems, suggests the creation of "**spike solutions**"—a design prototype
  - Encourages "**refactoring**"—an iterative refinement of the internal program design
- **XP Coding**
  - Recommends the **construction of a unit test** for a store before coding commences
  - Encourages "**pair programming**"
- **XP Testing**
  - All **unit tests are executed daily**
  - "**Acceptance tests**" are defined by the customer and executed to assess customer visible functionality

Extreme Programming (XP)



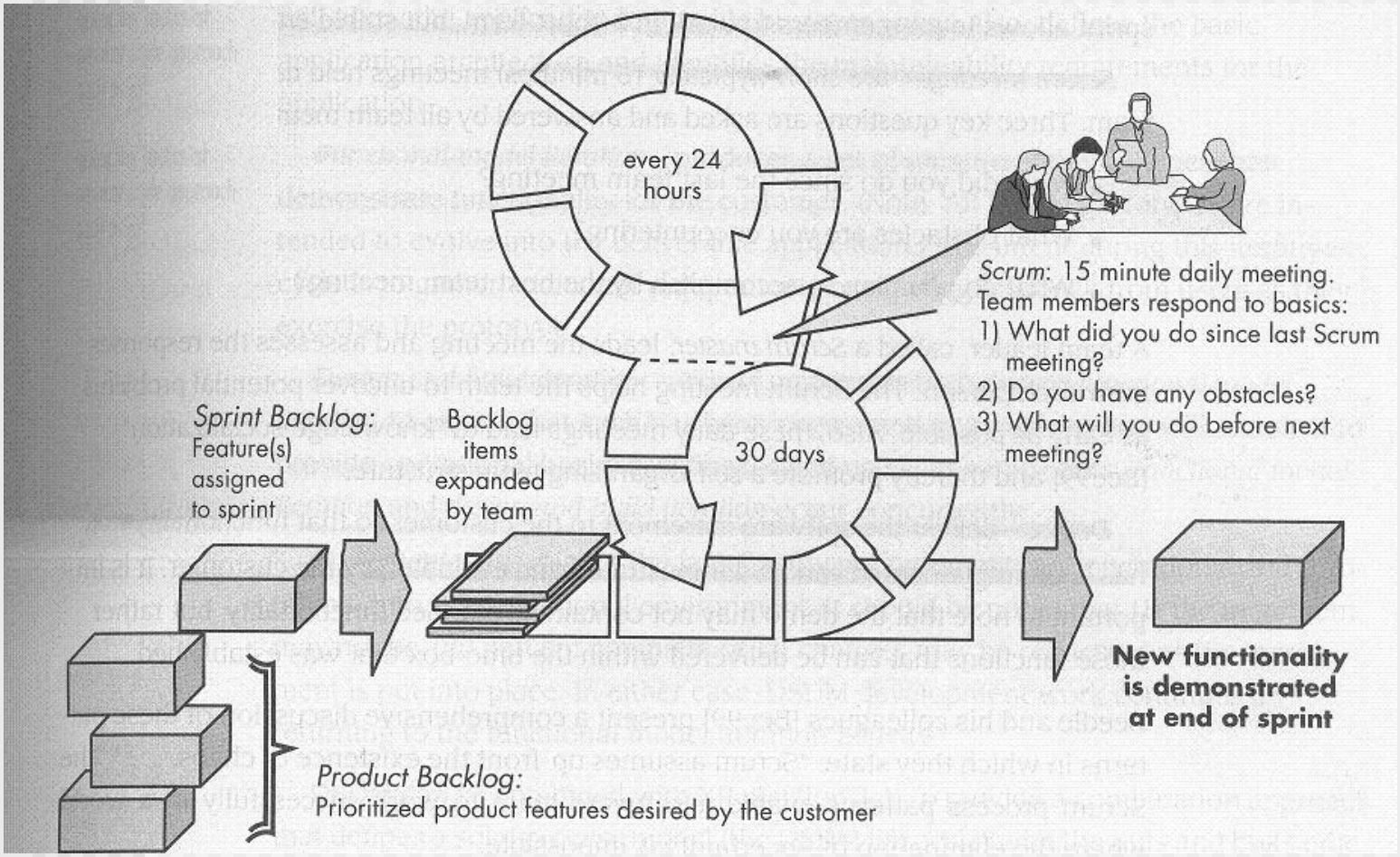
Industrial XP (IXP)

- IXP has greater inclusion of management, expanded customer roles, and upgraded technical practices
- IXP incorporates six new practices:
  - Readiness assessment
  - Project community
  - Project chartering
  - Test driven management
  - Retrospectives
  - Continuous learning

Scrum

- Originally proposed by Schwaber and Beedle.
- Scrum—distinguishing features:
  - Development work is partitioned into "**packets**".
  - **Testing and documentation** are on-going as the product is constructed.
  - Work occurs in "**sprints**" and is derived from a "**backlog**" of existing requirements.
  - **Meetings are very short** and sometimes conducted without chairs.
  - "**Demos**" are delivered to the customer with the time-box allocated.

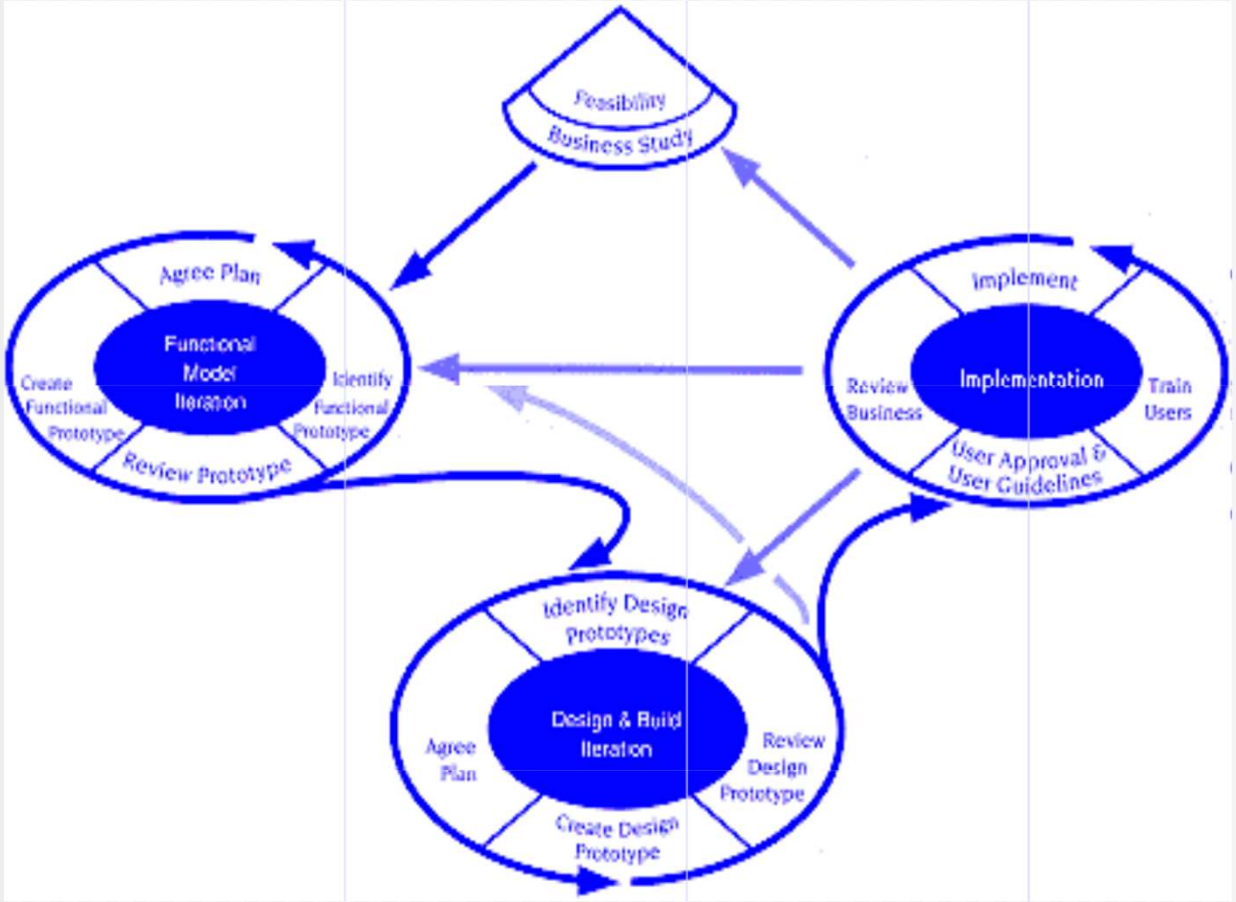
Scrum


every 24 hours
30 days
Scrum: 15 minute daily meeting.
Team members respond to basics:
1) What did you do since last Scrum meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Dynamic Systems Development Method

- Promoted by the DSDM Consortium (www.dsdm.org)
- DSDM—distinguishing features
  - Similar in most respects to XP
  - Nine guiding principles
    - Active user involvement is imperative.
    - DSDM teams must be empowered to make decisions.
    - The focus is on frequent delivery of products.
    - Fitness for business purpose is the essential criterion for acceptance of deliverables.
    - Iterative and incremental development is necessary to converge on an accurate business solution.
    - All changes during development are reversible.
    - Requirements are baselined at a high level
    - Testing is integrated throughout the life-cycle.

Dynamic Systems Development Method



DSDM Life Cycle (with permission of the DSDM consortium)

Agile Modeling

- Originally proposed by Scott Ambler
- Suggests a set of agile modeling principles
  - Model with a purpose
  - Use multiple models
  - Travel light
  - Content is more important than representation
  - Know the models and the tools you use to create them
  - Adapt locally

Agile Unified Process

- Each AUP iteration addresses these activities:
  - Modeling
  - Implementation
  - Testing
  - Deployment
  - Configuration and project management
  - Environment management