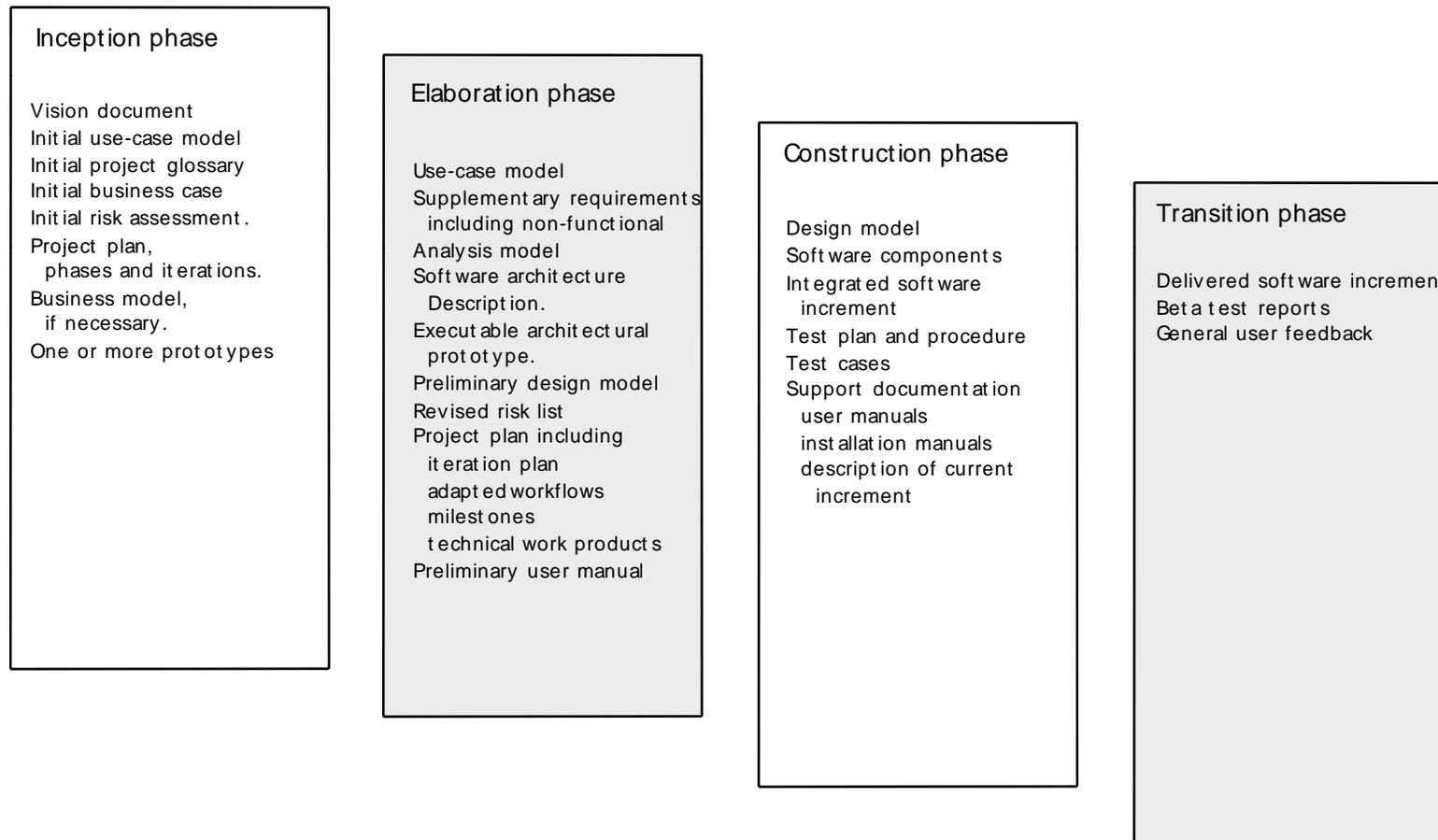


Chapter 4

Agile Development

Moonzoo Kim
CS Division of EECS Dept.
KAIST

Ex. UP Work Products



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** v.s. processes and tools
- **Working software** v.s. comprehensive documentation
- **Customer collaboration** v.s. contract negotiation
- **Responding to change** v.s. 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

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

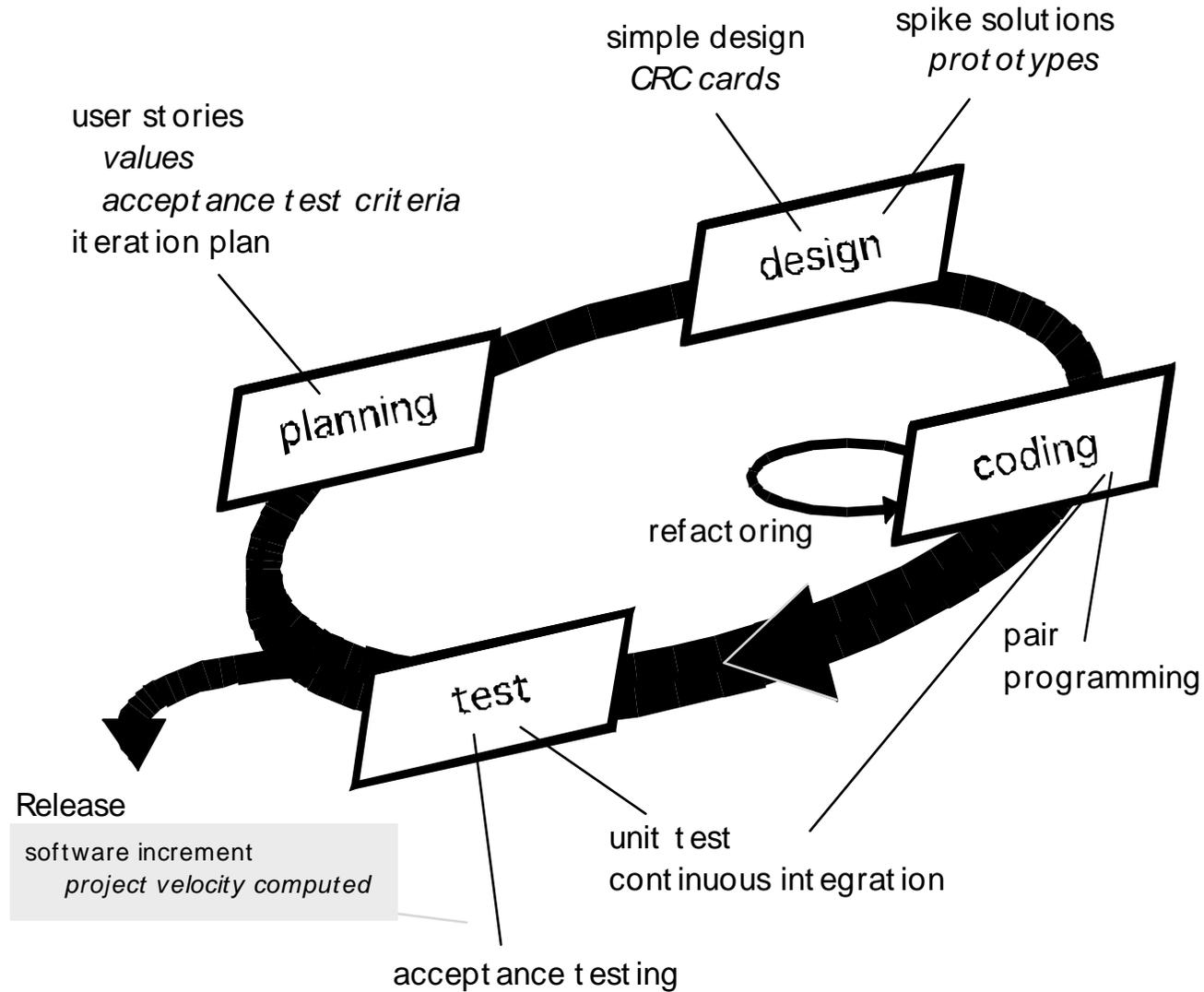
12 Principles of Agile Process

- To satisfy the customer through **early and continuous delivery** of valuable SW
- Welcome **changing requirements**, even late in development
- Deliver working SW **frequently** (from a couple of weeks to a couple of months)
- Business people and developers must **work together daily**
- Build projects around **motivated individuals**
- The most effective method of communication is **face-to-face** conversation
- **Working SW** is the primary measure of progress
- Agile processes promote sustainable development
- Continuous attention to technical excellence and good design
- Keep it simple (KIS)
- The best architectures, requirements, and designs emerge from **self-organizing teams**
- **At regular intervals**, the team reflects on how to become more effective, then tunes and adjusts its behavior

Extreme Programming (XP) (1/3)

- The most widely used agile process, originally proposed by Kent Beck
- XP Planning
 - Begins with the creation of “user stories” that describe required features and functionality for SW
 - Agile team assesses each story and assigns a cost (in weeks)
 - Stories are grouped to for a deliverable increment
 - A commitment is made on delivery date
 - After the first increment “project velocity” (# of user stories implemented during the first release) is used to help define subsequent delivery dates for other increments

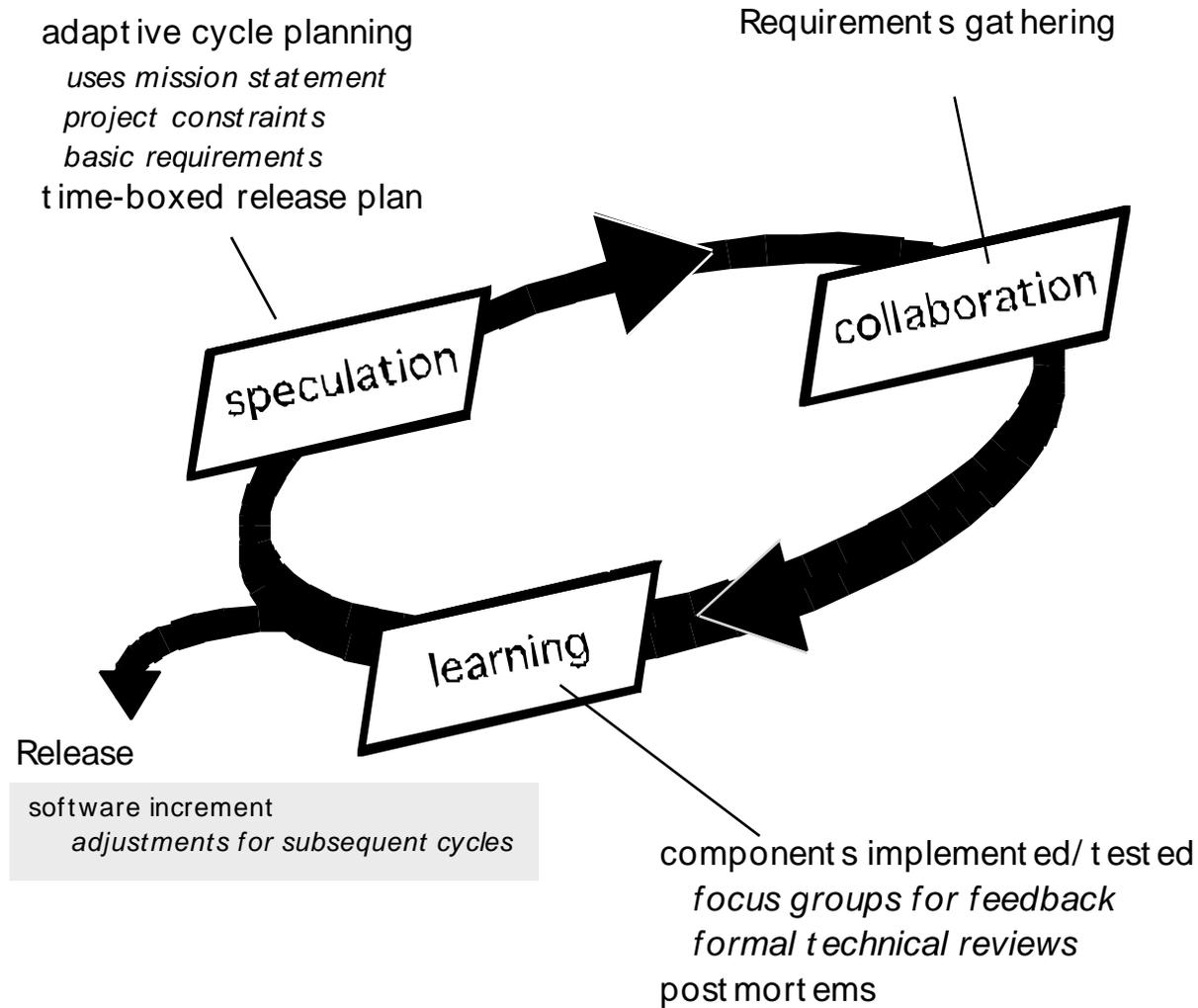
Extreme Programming (XP) (3/3)



Adaptive Software Development

- Originally proposed by Jim Highsmith
- Focus on human collaboration and team self-organization
- ASD — distinguishing features
 - Component-based focus
 - Explicit consideration of risks
 - Emphasizes collaboration for requirements gathering
 - Emphasizes “learning” throughout the process

Adaptive Software Development



Dynamic Systems Development Method

- Promoted by the DSDM Consortium (www.dsdm.org)
- DSDM—distinguishing features
 - Similar in most respects to XP and/or ASD
 - 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.

Scrum

- Originally proposed by Schwaber and Beedle
- Scrum—distinguishing features
 - Testing and documentation are on-going as the product is constructed
 - Work occurs in “sprints” and is derived from a “backlog” of existing requirements
 - Backlog: a prioritized list of project requirements or features
 - Sprint: work tasks within a process pattern
 - Daily meetings are very short and sometimes conducted without chairs
 - Demos are delivered to the customer within the time-box allocated

Feature Driven Development

- Originally proposed by Peter Coad et al
- FDD—distinguishing features
 - Emphasis is on defining “features”
 - a *feature* “is a client-valued function that can be implemented in two weeks or less.”
 - Users can describe features more easily
 - Uses a feature template
 - `<action> the <result> <by | for | of | to> a(n) <object>`
 - Ex. Add the product to a shopping cart
 - Ex. Store the shipping-information for a customer
 - Feature set template
 - `<action> <-ing> a(n) <object>`
 - Ex. Making a product sale
 - A features list is created and “plan by feature” is conducted
 - Design and construction merge in FDD

Agile Modeling

- Originally proposed by Scott Ambler
- Suggests a set of agile modeling principles for building large, business critical systems
 - 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

HW #2: Due Mar 3rd 23:59

- Select and read any two articles on agility in June 2003 issue of IEEE Computer
 - Agile Software Development: It's about Feedback and Change
 - Williams, L.; Cockburn, A.
 - Agility through discipline: a debate
 - Beck, K.; Boehm, B.
 - Migrating Agile Methods to Standardized Development Practice
 - Lycett, M.; Macredie, R.D.; Patel, C.; Paul, R.J.
 - Using Risk to Balance Agile and Plan-Driven Methods
 - Boehm, B.; Turner, R.
 - Etc.
 - <http://ieeexplore.ieee.org/>
- Summarized those two articles you selected with your opinion in one A4 each (around 400 words)
 - Be careful to write in a literary style, not conversational style