# Chapter 4
# Agile Development

**Software Engineering: A Practitioner's Approach, 6th edition**
*by Roger S. Pressman*

## 本章要点

- 敏捷开发
  - ➢极限编程（XP）
  - ➢自适应软件开发(ASD)
  - ➢动态系统开发方法 (DSDM)
  - ➢Scrum模型
  - ➢Crystal模型
  - ➢特征驱动开发（FDD）

## Common Fears for Developers

- The project will produce the wrong product[有错的产品].
- The project will produce a product of inferior quality[低质量的产品].
- The project will be late[延迟].
- We’ll have to work 80 hour weeks[每周工作80小时].
- We’ll have to break commitments[违约].
- We won’t be having fun[没有休闲时间].

# The Manifesto for Agile Software Development

**2001年Kent Beck和其它16位知名软件开发者、软件工程专家以及软件咨询师(称为敏捷联盟)共同签署"敏捷软件开发宣言"，该宣言声明:**

"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 <u>processes and tools</u>
- Working software over <u>comprehensive documentation</u>
- Customer collaboration over <u>contract negotiation</u>
- Responding to change over <u>following a plan</u>

That is, while there is value in the items on the right, we value the items on the left more."[虽说上述右边的各项很有价值,但我们认为左边的各项具有更大的价值]

**-- Kent Beck et al(2001).**

## 敏捷开发是一场运动

- 本质上讲，敏捷方法是为了克服传统软件工程中认识和实践的弱点设计而成的。敏捷开发带来多方面的好处，但它不适用于所有的项目、所有的方面、所有的人和所有的情况，它并不独立于传统的软件工程实践，也不能作为超越一切的哲学理念而用于所有软件工作。

## What is "Agility"?

- *适应变更*: Effective (rapid and adaptive) response to change
- *交流通畅*: Effective communication among all stakeholders
- *客户参与*: Drawing the customer onto the team[吸收]
- *有效控制*: Organizing a team so that it is in control of the work performed

*Yielding …*

- Rapid, incremental delivery of software

## 12 Principles of Agility——敏捷联盟[2003]

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.[我们最先要做的是通过尽早、持续交付有价值的软件来使客户满意]
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage. [即使在开发的后期，也欢迎需求变更。敏捷过程利用变更为客户创造竞争优势]
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale. [经常交付可工作软件,交付的时间间隔可以从几个星期到几个月,交付的时间间隔越短越好]
4. Business people and developers must work together daily throughout the project.[在整个项目开发期间,业务人员和开发人员必须天天都在一起工作]

## Principles of Agility…

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.[围绕受激励的个人构建项目.给他们提供所需的环境和支持,并且信任他们能够完成工作]
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.[在团队内部,最富有效果和效率的信息传递方法是面对面交谈]
7. Working software is the primary measure of progress. [可工作软件是进度的首要度量指标]
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.[敏捷过程提倡可持续的开发速度.赞助人、开发者和用户应该能够保持一种长期的、稳定的开发速度]

## Principles of Agility…

9. Continuous attention to technical excellence and good design enhances agility. [不断地关注优秀的技能和好的设计会增强敏捷能力]
10. Simplicity - the art of maximizing the amount of work not done - is essential. [简单—使不需要的工作最大化的艺术—是必要的]
11. The best architectures, requirements, and designs emerge from self-organizing teams. [好的架构、需求和设计出自于自组织团队]
12. At regular intervals, the team reflects on[反省] how to become more effective, then tunes and adjusts its behavior accordingly. [每隔一定时间，团队会反省如何才能更有效地工作，并相应调整自己的行为]

## An Agile Process

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

# Extreme Programming (XP)

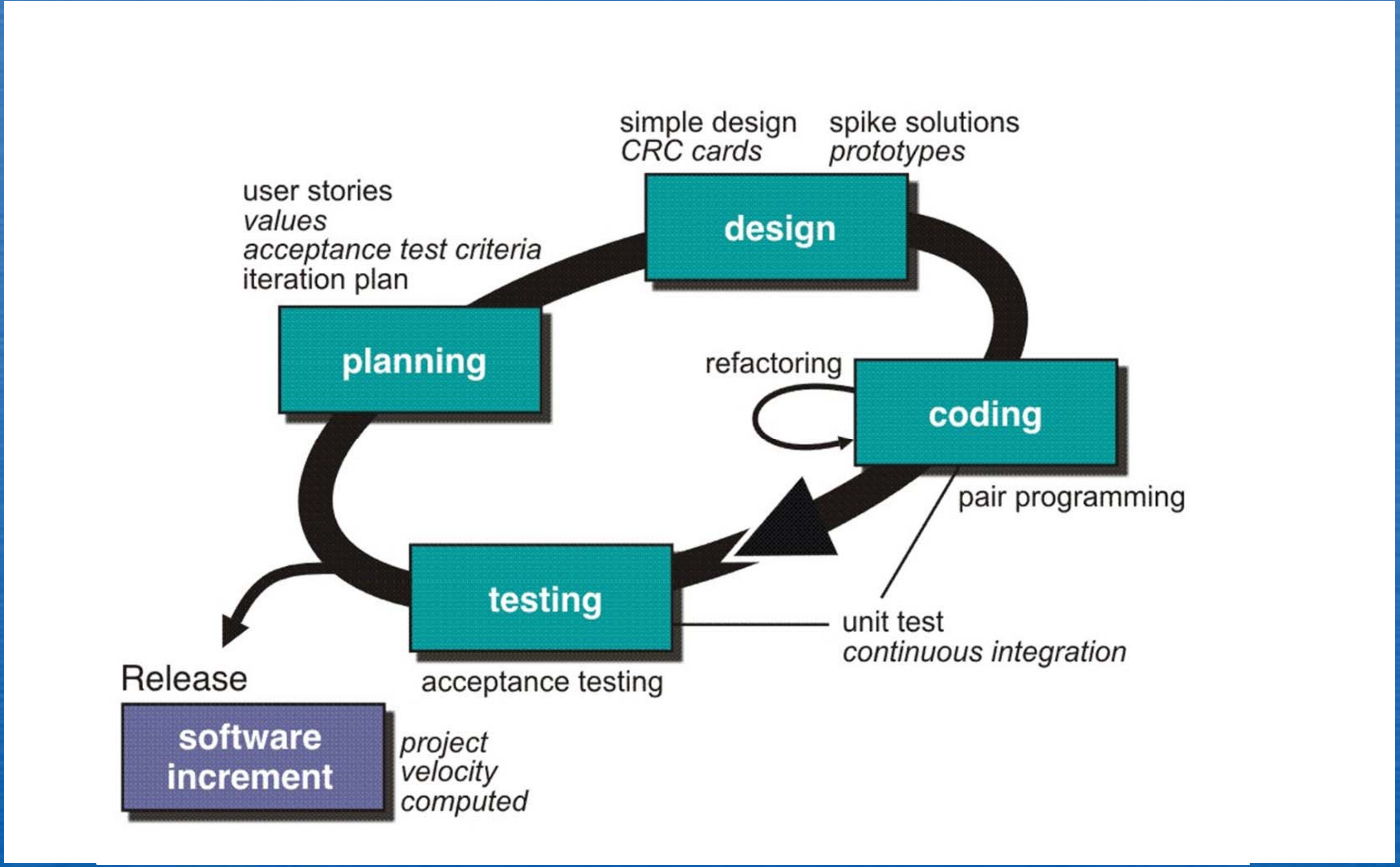
- The most widely used agile process, originally proposed by Kent Beck[1999]
- XP Planning
  - Begins with the creation of user stories
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment[承担义务] is made on delivery date[交货日期]
  - 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 cards (see Chapter 8)
  - 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 story *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

立即建立这部分设计的可执行原型，实现并评估设计原型

# Extreme Programming (XP)...



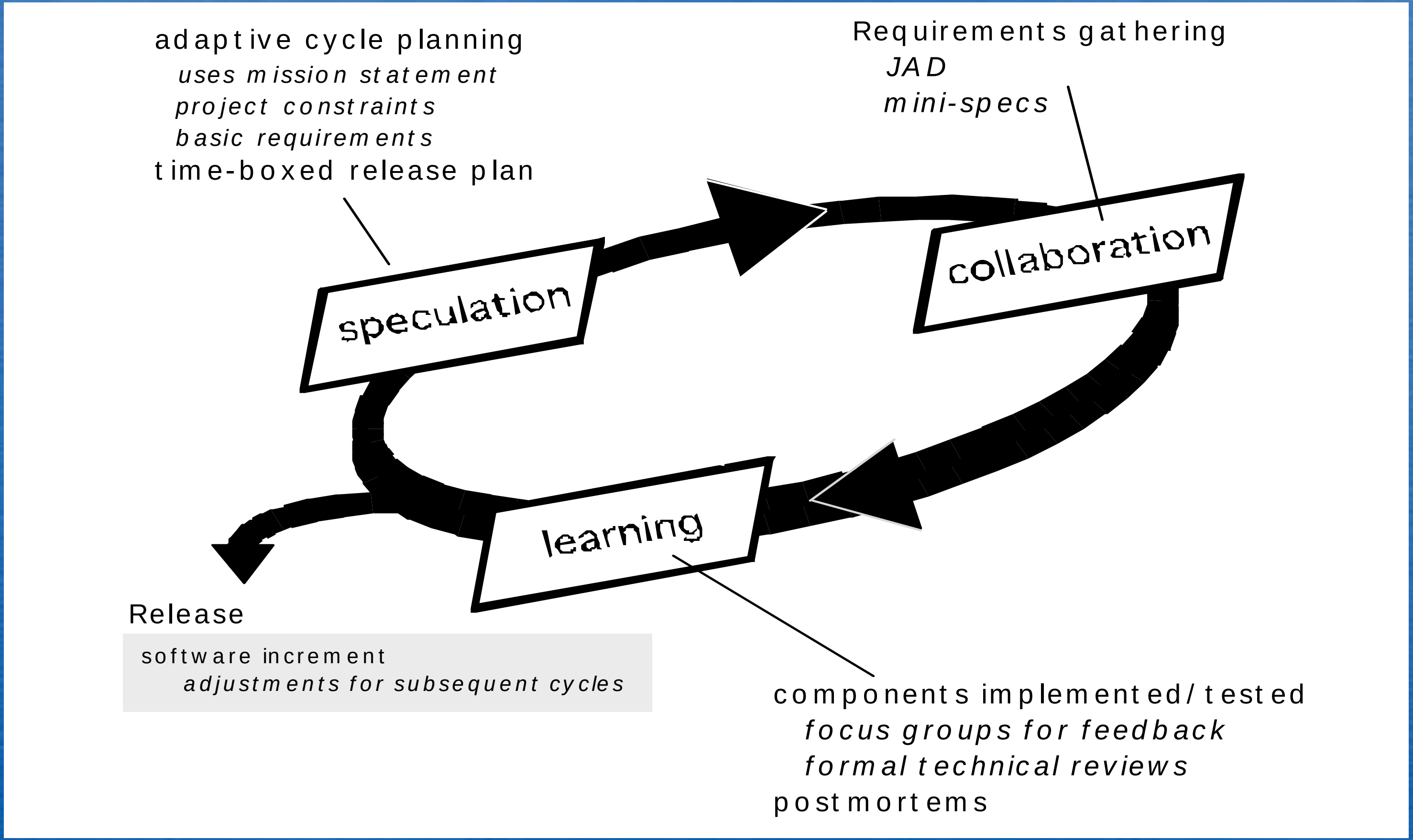
## Other Agile Processes

- Adaptive Software Development (ASD)
- Scrum
- Feature Driven Development
- ……

## Adaptive Software Development

- Originally proposed by Jim Highsmith[2000]
- ASD — distinguishing  features
  - Mission-driven planning
  - Component-based focus
  - Uses “time-boxing” (See Chapter 24)
  - Explicit consideration of risks
  - Emphasizes collaboration for requirements gathering
  - Emphasizes “learning” throughout the process

# Adaptive Software Development...



adaptive cycle planning
*uses mission statement*
*project constraints*
*basic requirements*
time-boxed release plan

Requirements gathering
*JAD*
*mini-specs*

Release
software increment
*adjustments for subsequent cycles*

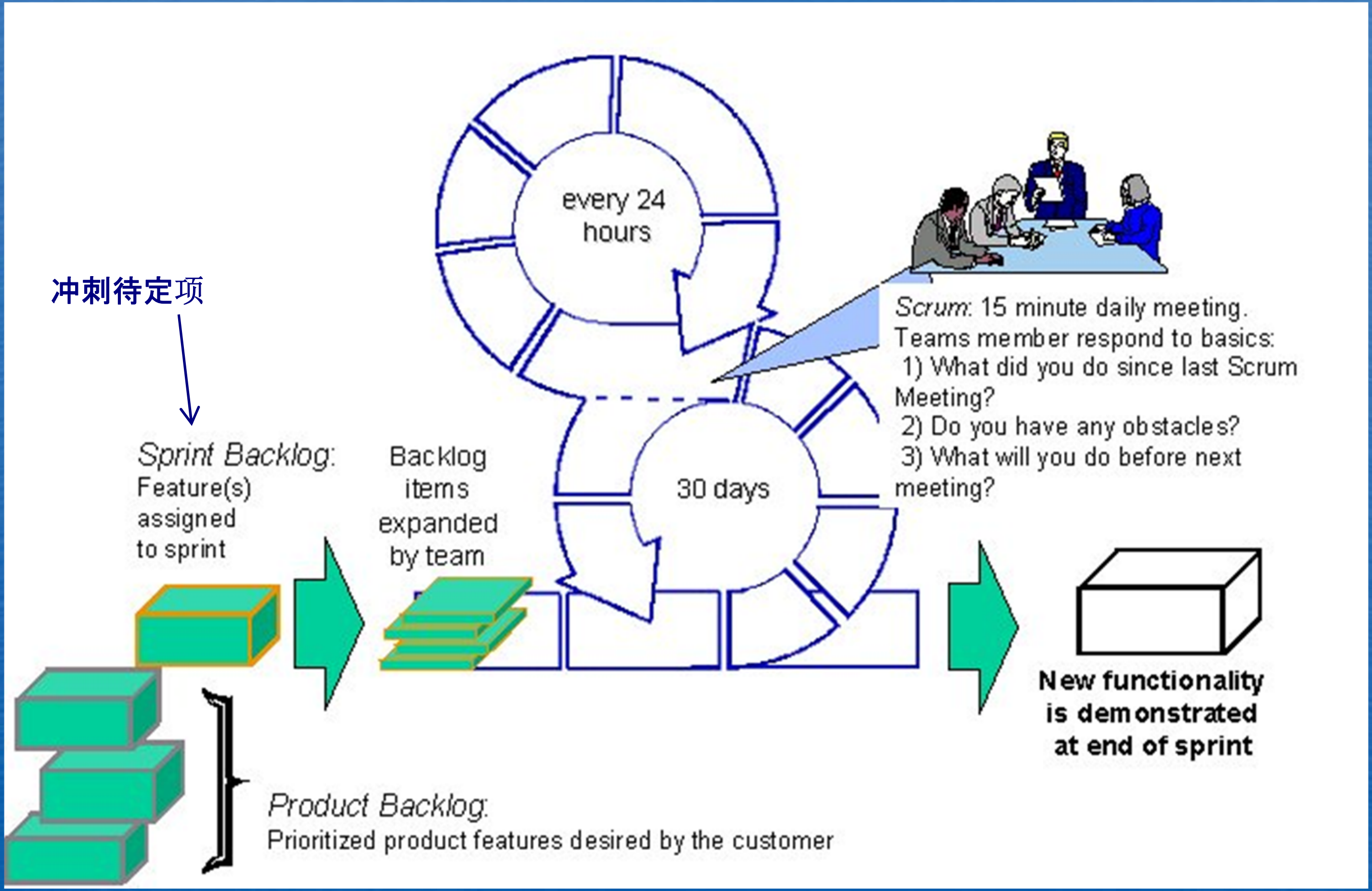
components implemented/tested
*focus groups for feedback*
*formal technical reviews*
postmortems

Joint Application Development (JAD)

## Scrum（争球）

- Originally proposed by Schwaber and Beedle[2001]
- Scrum—distinguishing features
  - Development work is partitioned into “packets”
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in “sprints[冲刺]” and is derived from a “backlog[待定项]” of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - “demos” are delivered to the customer with the time-box allocated

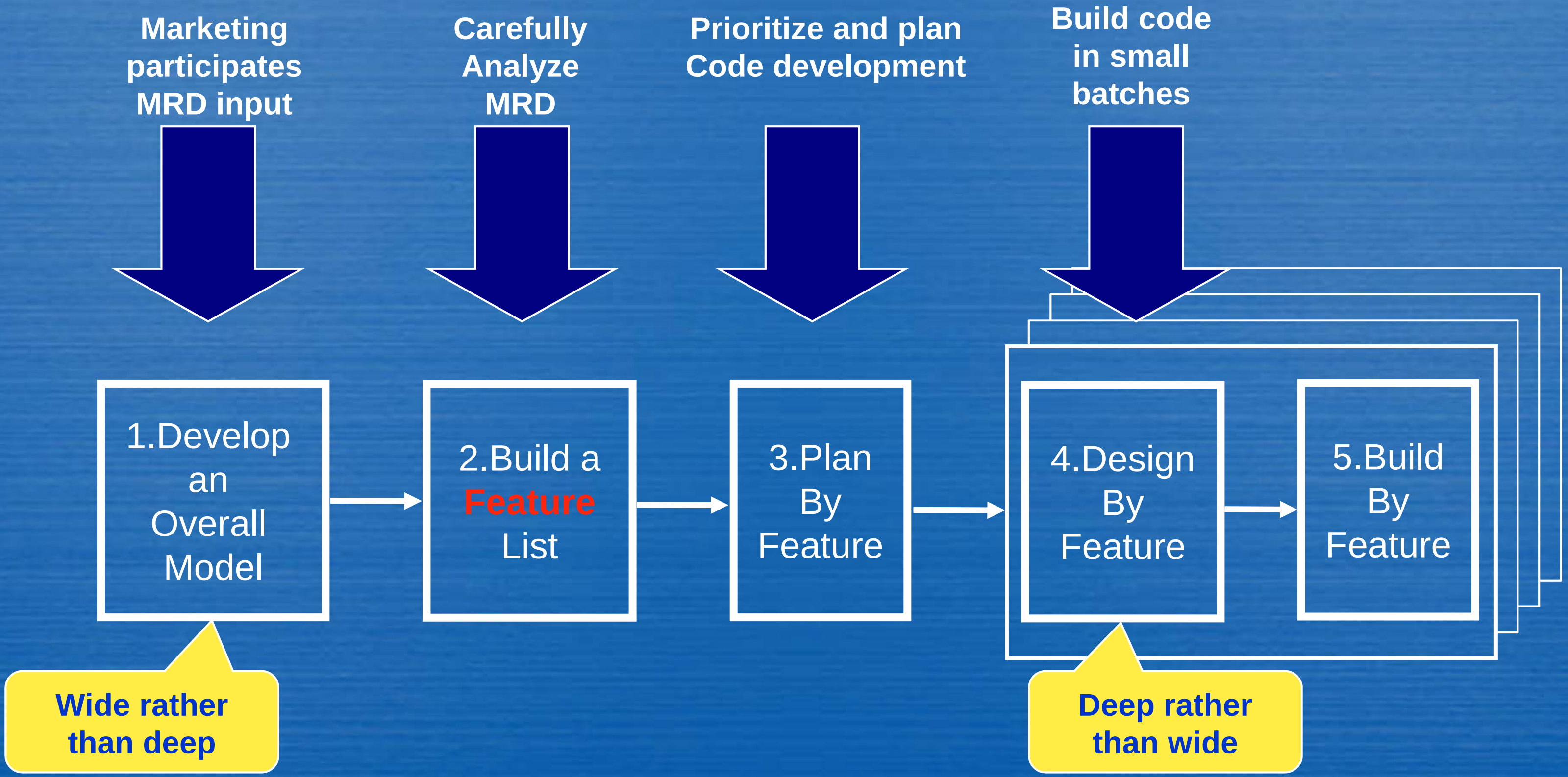
# Scrum…


冲刺待定项
every 24 hours
30 days
Sprint Backlog: Feature(s) assigned to sprint
Backlog items expanded by team
Scrum: 15 minute daily meeting. Teams member 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?
New functionality is demonstrated at end of sprint
Product Backlog: Prioritized product features desired by the customer

## Feature Driven Development

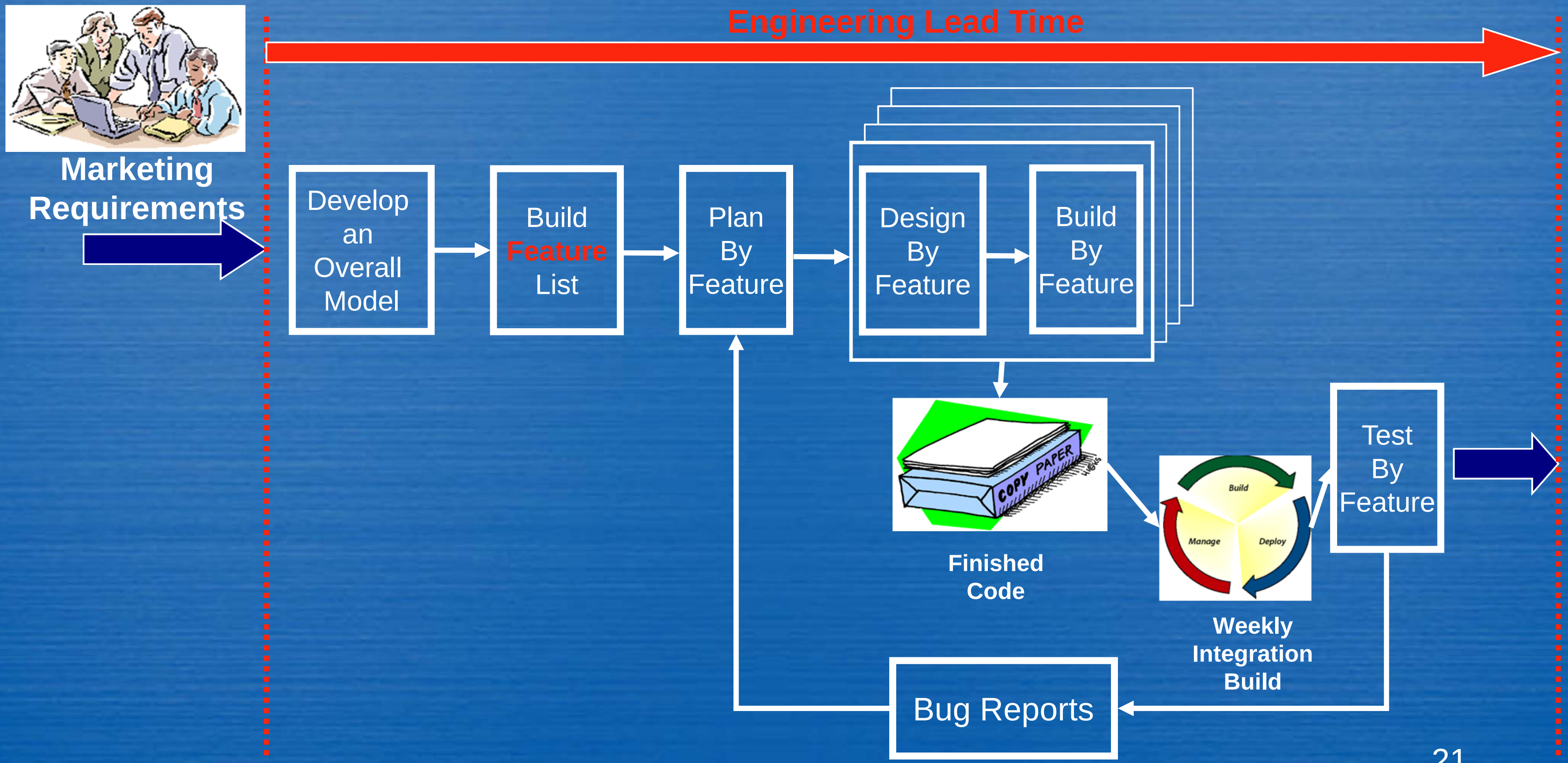
- Originally proposed by Peter Coad et al[1999]
- FDD—distinguishing features
  - Emphasis is on defining "features"
    - a feature "is a client-valued function that can be implemented in two weeks or less."
  - Uses a feature template
    - <action> the <result> <by | for | of | to> a (n) <object>
    - For example, Add the technical-specifications of a product
  - A features list is created and "plan by feature" is conducted
  - Design and construction merge in FDD

## Feature Driven Development…



MRD: Market Requirement Document

## Engineering process


Engineering Lead Time
Marketing Requirements
Develop an Overall Model
Build Feature List
Plan By Feature
Design By Feature
Build By Feature
Finished Code
Build
Manage
Deploy
Weekly Integration Build
Test By Feature
Bug Reports