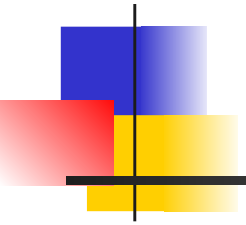
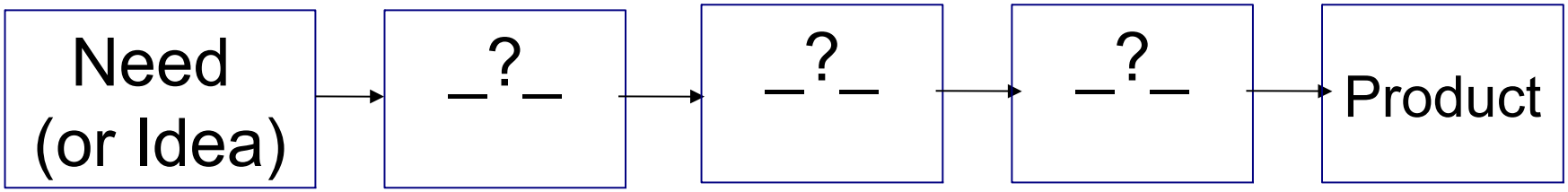


Intro to Software Processes

Goal of Software Development

Produce a software product that fulfills a need or realizes an idea.

What are the Steps?



What are the major steps or activities you would need to do?

List as many as you can.

Software Development Steps

1.
2.
3.
4.
5.

Activities in Software Development

- elicit requirements
- specification
- high-level design
- prototyping (maybe)
- detailed (software) design
- construction & testing
- validation
- documentation
- transition
- maintenance

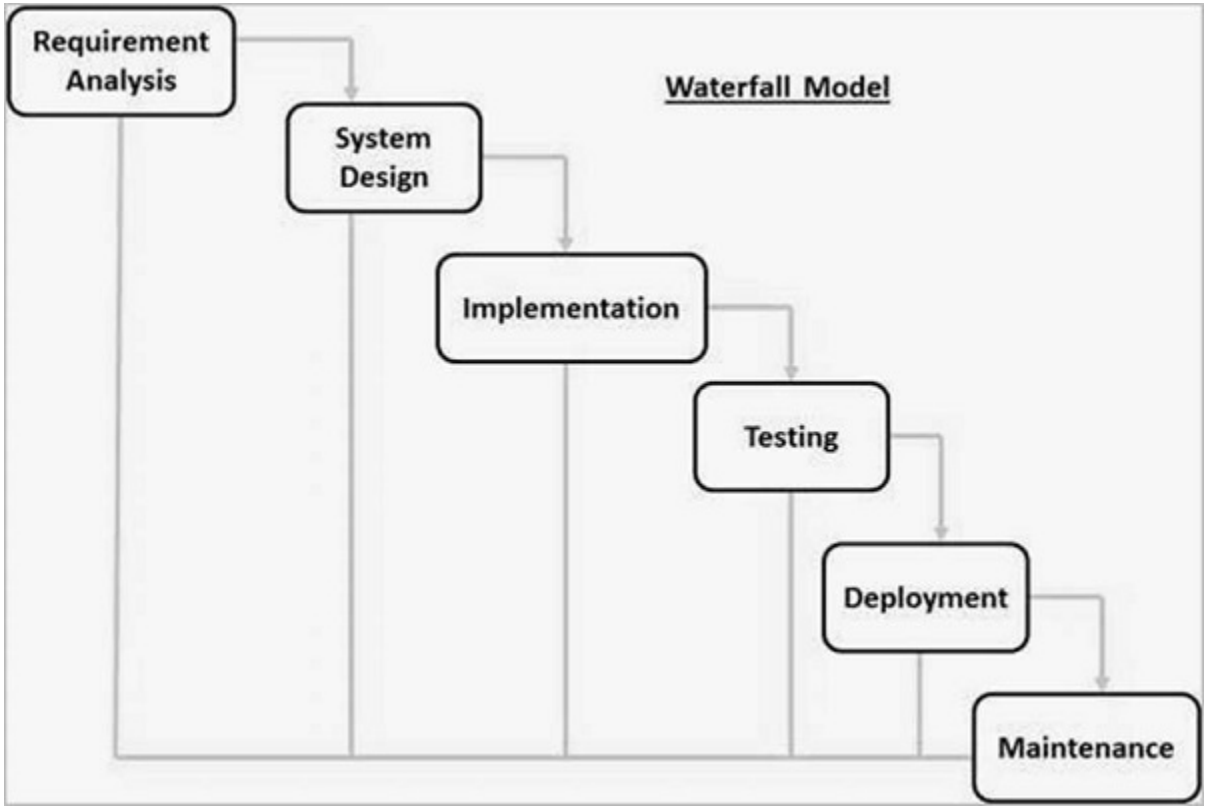
Managing the project involves

- planning
- obtaining resources
- measuring
- tracking progress
- review
- analyze results, take action as needed

How about if we do them in order?

This is a logical way to proceed.

Similar to a civil engineering project.



(There may be more steps than this.)

What Could Go Wrong?

Consider these cases

What would be effect on project of each of these?

1. You fail to elicit some requirement(s).

2. You misunderstand some requirements, so the design is not what the customer wanted.

3. The framework you chose for the software can't handle requirements.

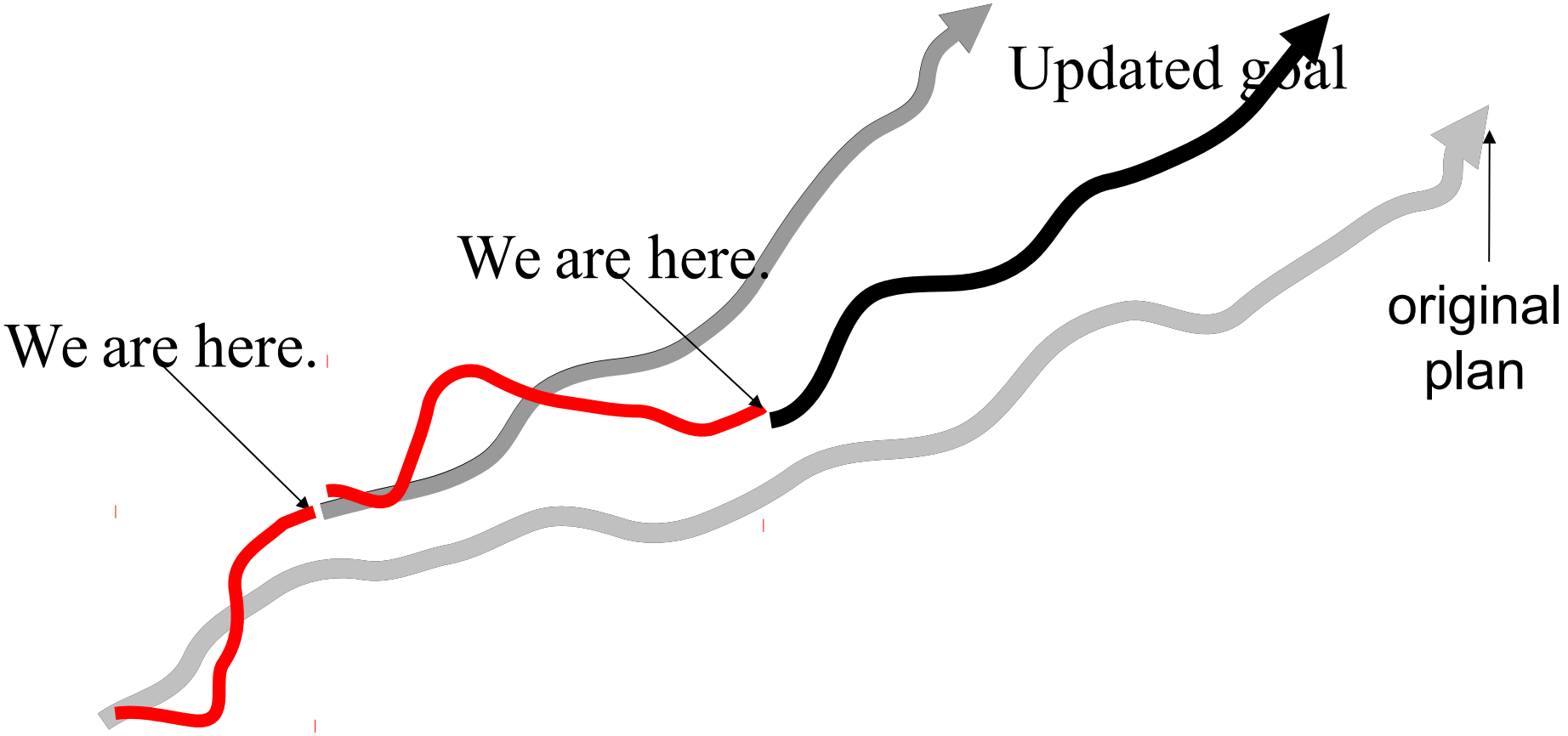
4. Lots of bugs during development, discovered during test phase.

How to Avoid These Problems?

Frequently "check" your progress to see if you are on the right track.

- Feedback
- Testing
- Review
- Analyze results and take corrective action

Using Feedback



Overview of some Process Models

"Model" of how software development (SDLC) should go.

A "model" is an abstraction of something else.

So, some details may be missing and it may be imprecise.

Code and Fix

- The most common software development process
- *ad hoc* (chaos). Little or no planning and design.

Code and Fix

1. think about the problem, doodle ideas on paper
2. start coding
3. run it. Test what I just wrote and fix it.
4. as code grows, I realize that I need to change some parts.
   - modify the code
   - goto step 2.

Why "Code and Fix" is Inefficient

- No clear statement of what the result should be

- No design or "its in my head".

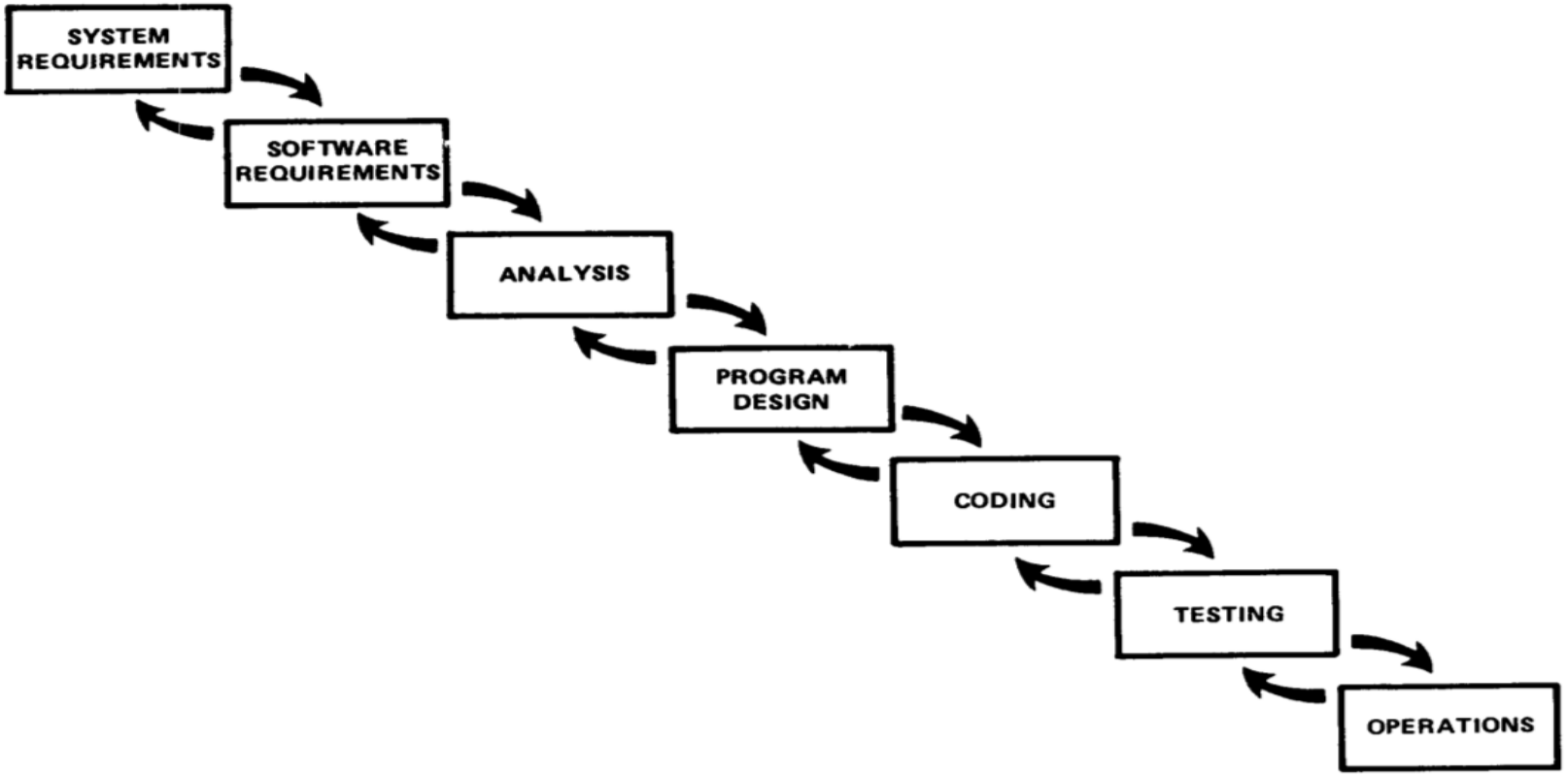
- Lots of rework.

Waterfall Model

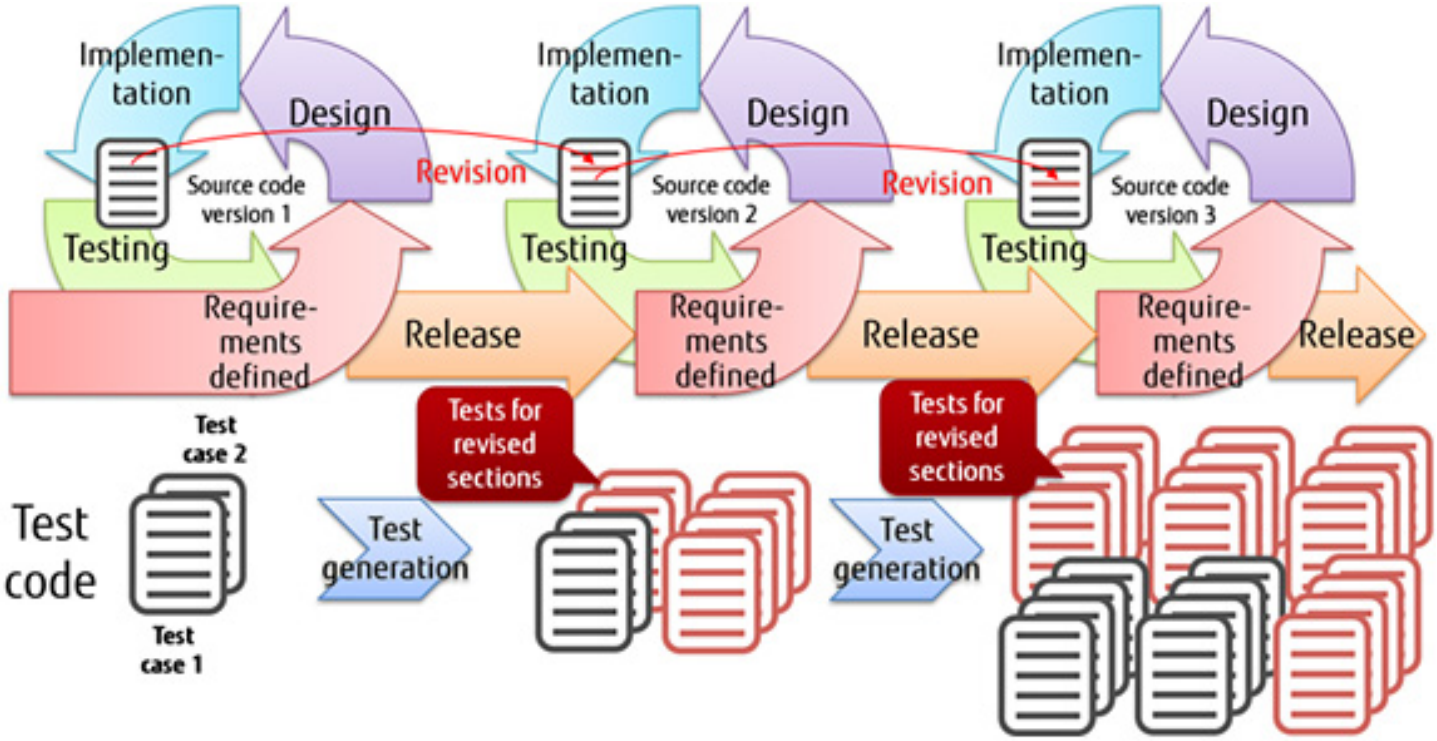
Winston Royce, *Managing the Development of Large Software Systems* (1970)

Still commonly used.

Not as simple as usually portrayed.



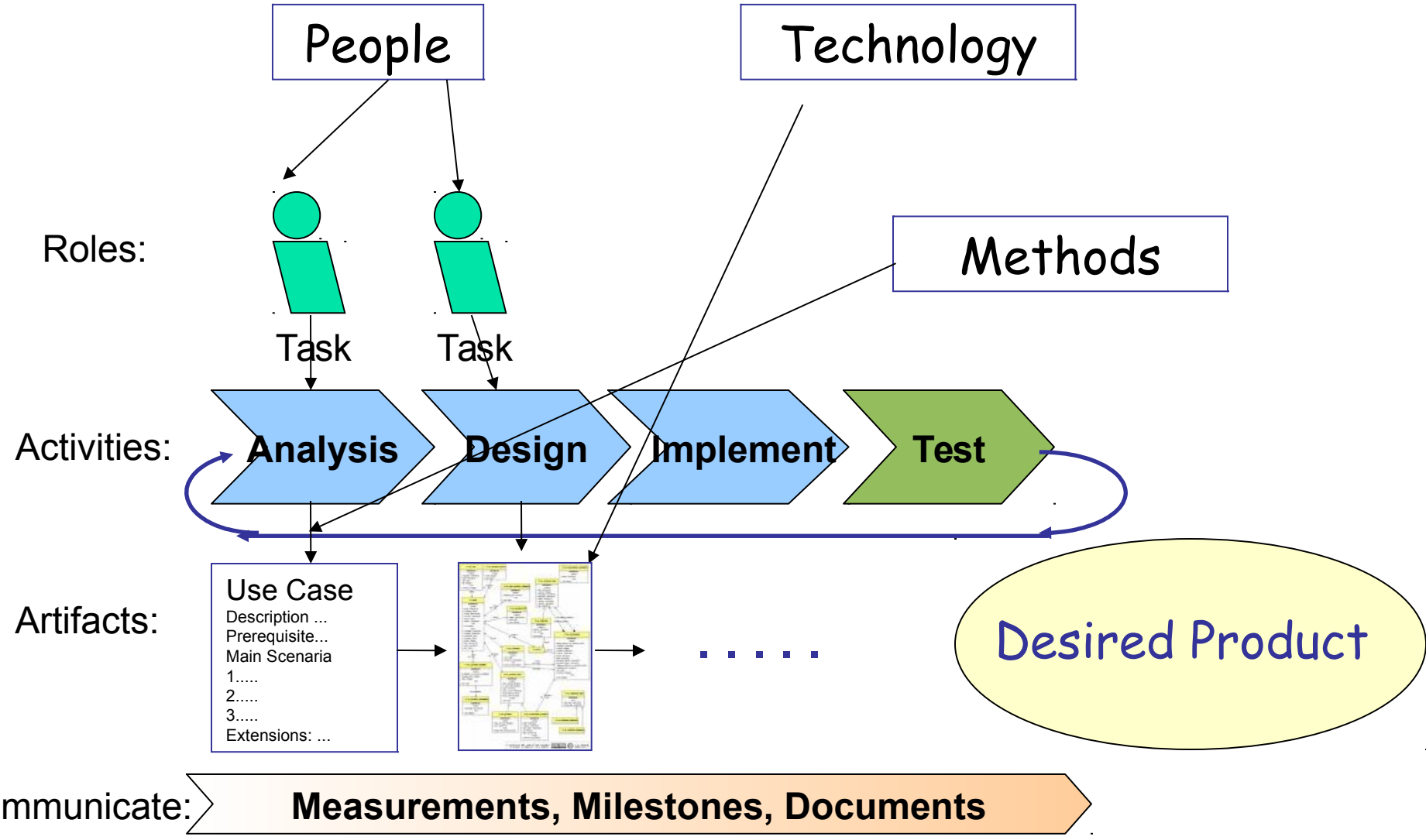
Iterative and Incremental



Incremental - project broken down into increments containing some useful features and/or other products.

Iterative - repeat it until satisfactory product is delivered.

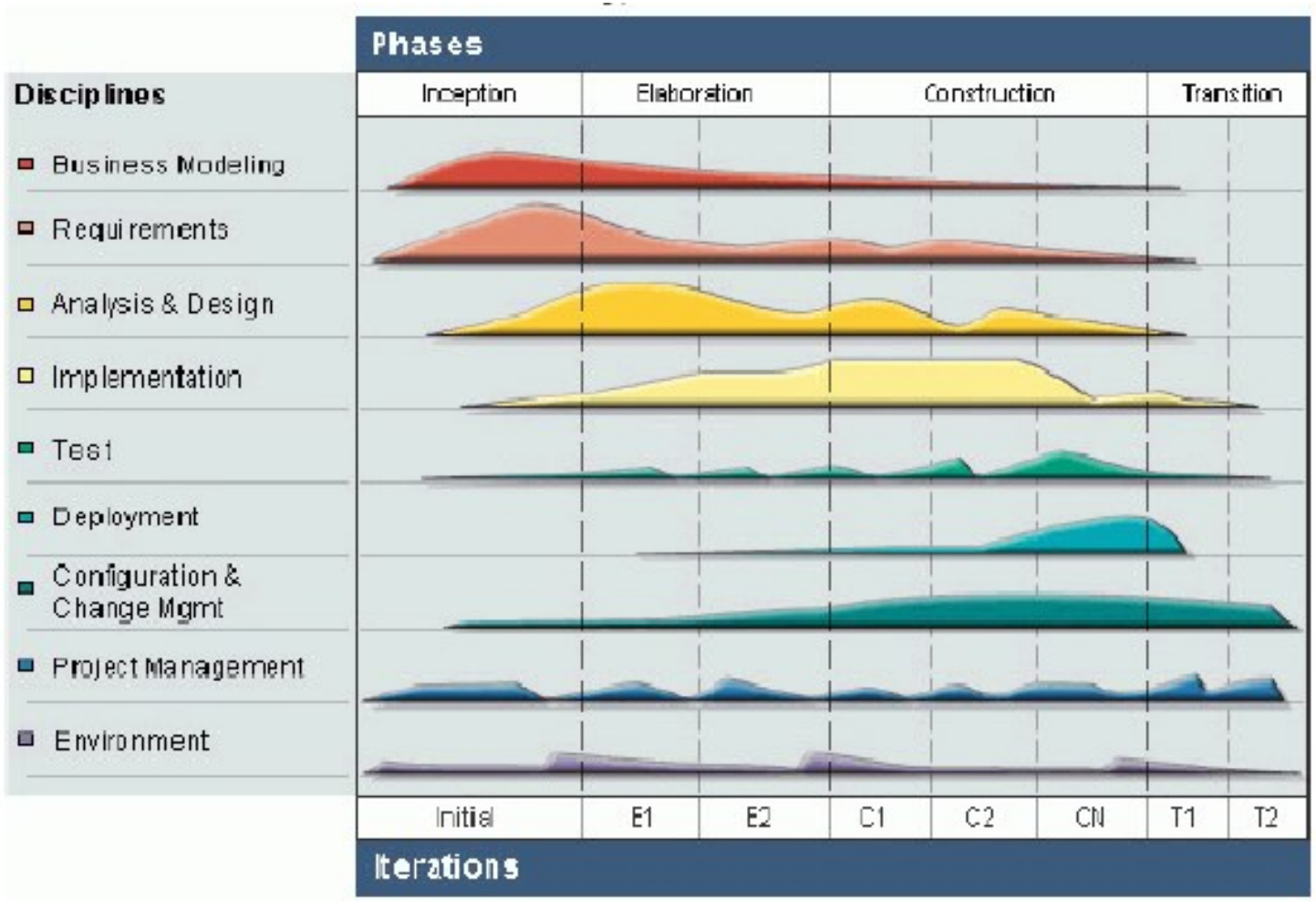
Unified Process Model



UP Disciplines and Workflow

Workflows related to different areas of concern.

Workflow contains activities and tasks.



Unified Process Characteristics

- Characteristics
  - time-boxed iterations
  - plan based
  - architecture centric
  - address risks early
  - implement requirements based on priority or business value
  - accommodate change
- Unified Software Development Process - a framework
  - Rational UP
  - Open-UP
  - Agile UP (Scott Amber)

Agile Manifesto

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

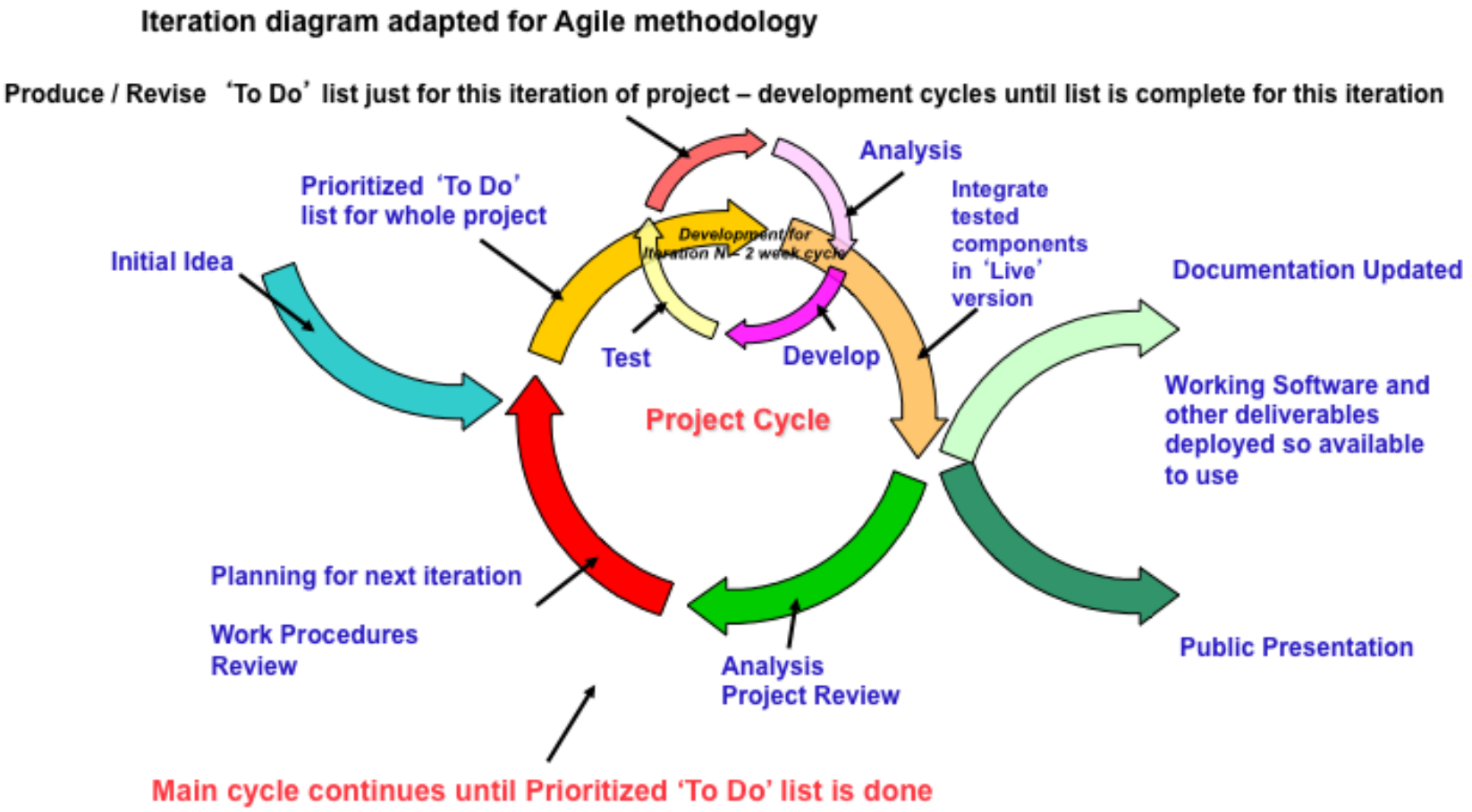
Customer collaboration over contract negotiation

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.

Agile Characteristics

See: *12 Agile Principles and Practices*



Agile Process Characteristics

- create customer "value" at each iteration
- welcome evolving requirements
- working software as primary measure of progress
- lack of up-front architecture design (YAGNI)
- simplicity of design (XP: "*do the simplest thing that* ...")
- small, self-organizing teams at one site (preferred)
- frequent customer feedback
- shared understanding instead of *comprehensive* documents … but they *do* write documents

...

Some Agile Processes

- eXtreme Programming
  - Kent Beck: Chrysler
- Scrum - called "more a management technique"
  - iterative development in "Sprints"
  - daily stand-up meeting
  - small, democratic teams
- Crystal
  - a family of methods to address different types of projects
- Synchronize and Stabilize (Microsoft process)

What About Individual Process?

Many software processes. *So what?*

... This is a course about individual process.

Can you define:

1. Goals for an I.S.P.?
2. Key practices and skills, based on the goals?