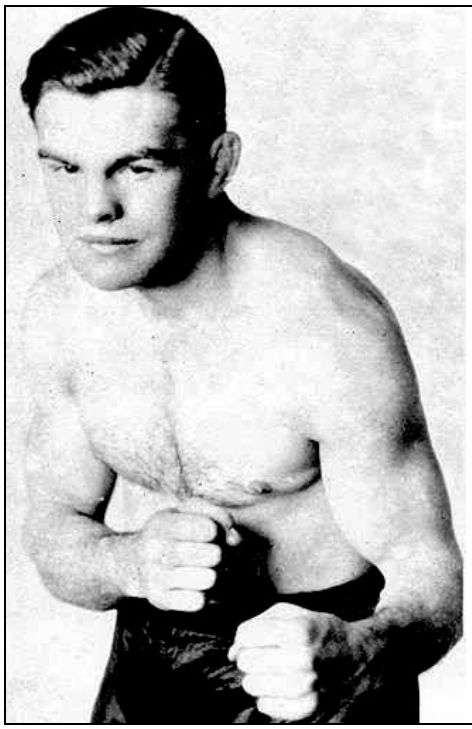


An Agile UP: Introduction

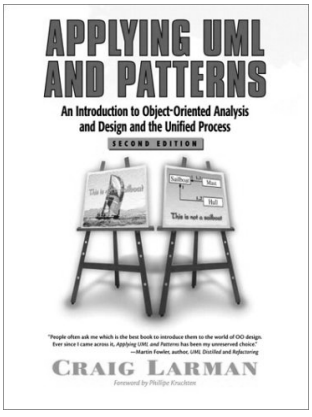
Craig Larman

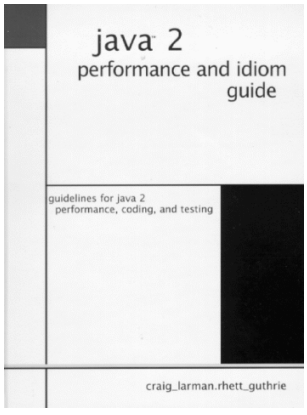
www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

speaker background

- www.craiglarman.com
- Roles at Intell*Agile* & Valtech. . .



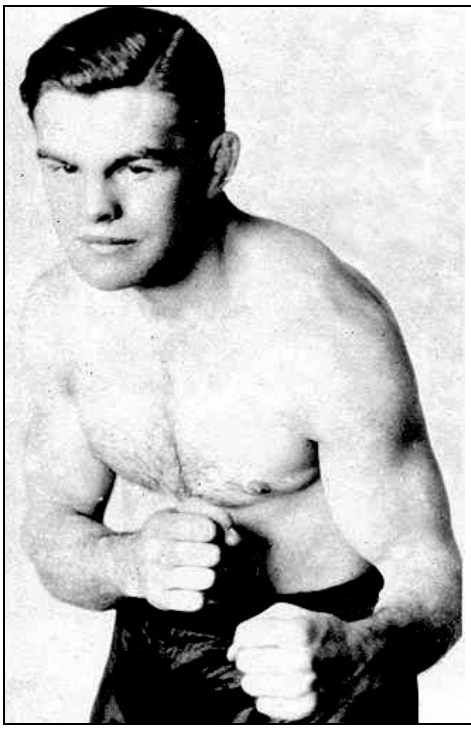


Underway:

Agile and Iterative Development: A Manager's Guide

2

Copyright © Craig Larman. www.craiglarman.com



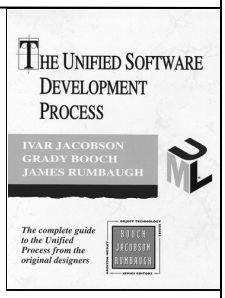
UP Overview

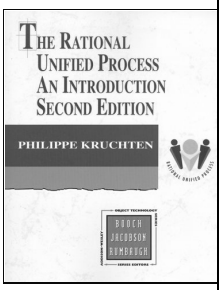
www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

Defining our Terms: UP and RUP

- The RUP *process framework* is a detailed, updated refinement of the more general UP process framework.
- RUP lead architect: Philippe Kruchten.
- Some use the term “UP” for “UP family” — those related to (and consistent with) the UP process framework, especially the RUP.

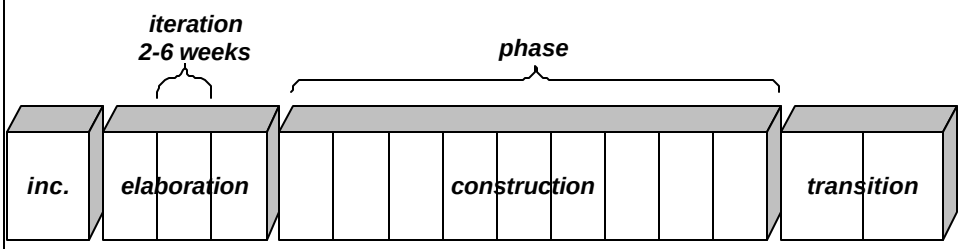




4

Copyright © Craig Larman. www.craiglarman.com

UP Practice #1: Develop Iteratively

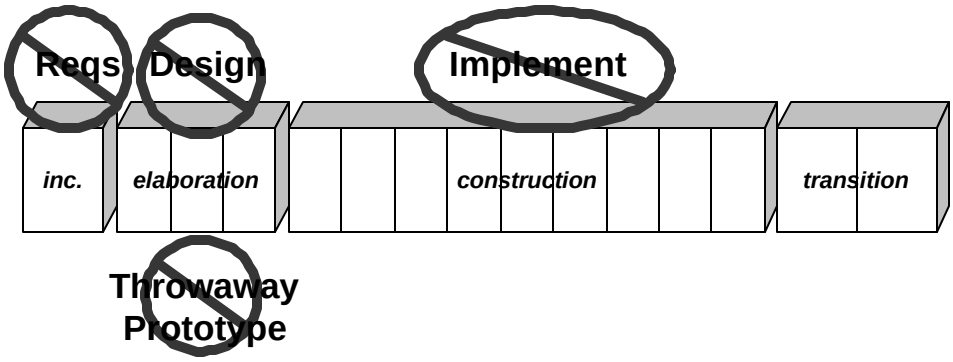


Work in early iterations emphasize:

- **Risk**
- **Criticality (high business value)**
- **Coverage (many components are touched)**

5 Copyright © Craig Larman. www.craiglarman.com

UP Practice #1: Develop Iteratively



6 Copyright © Craig Larman. www.craiglarman.com

UP Practice #2: Continuously Verify Quality



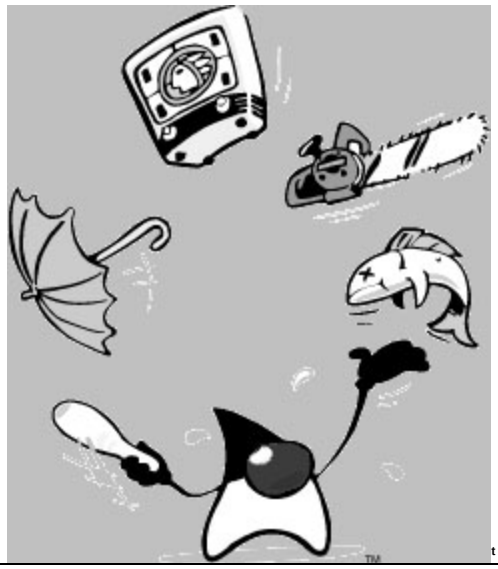
7

UP Practice #3: Build a Cohesive Architecture



Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

UP Practice #4 and #5:
Manage Requirements and Manage Change



9

t © Craig Larman. www.craiglarman.com

UP Practice #6:
Model Visually

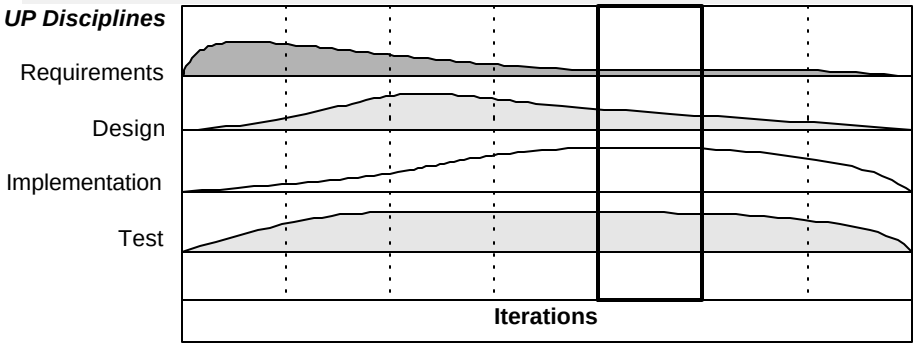




10

Copyright © Craig Larman. www.craiglarman.com

Other Key UP Ideas



Deployment Discipline *Optional* Artifacts



11

w.craiglarman.com

Other Key UP Ideas

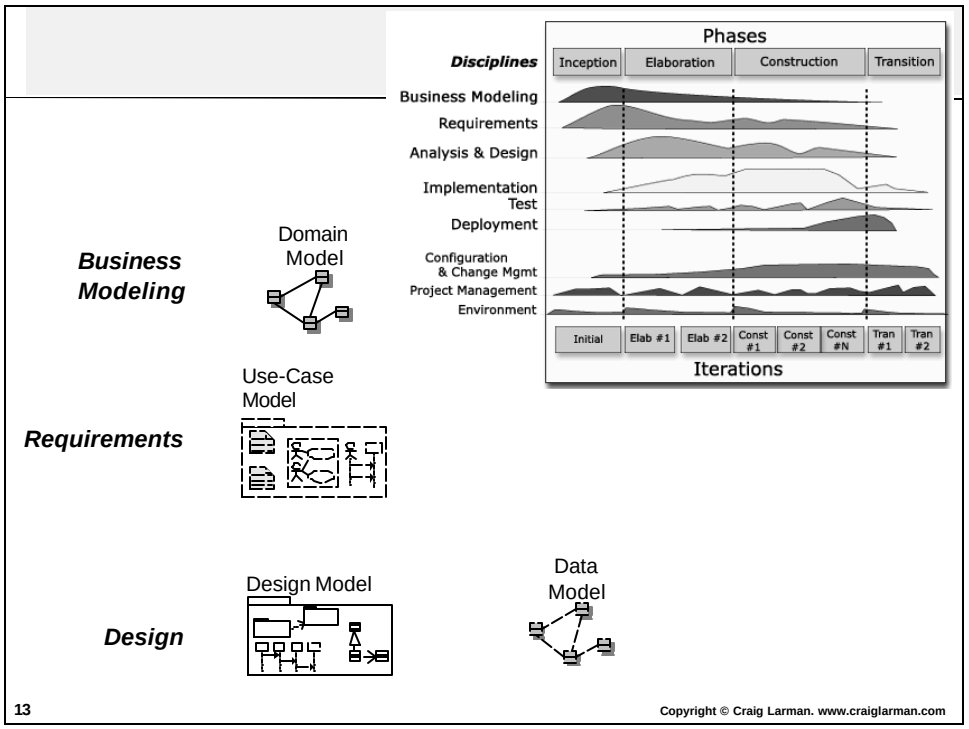
Sample Development Case (s-start; r-refine)

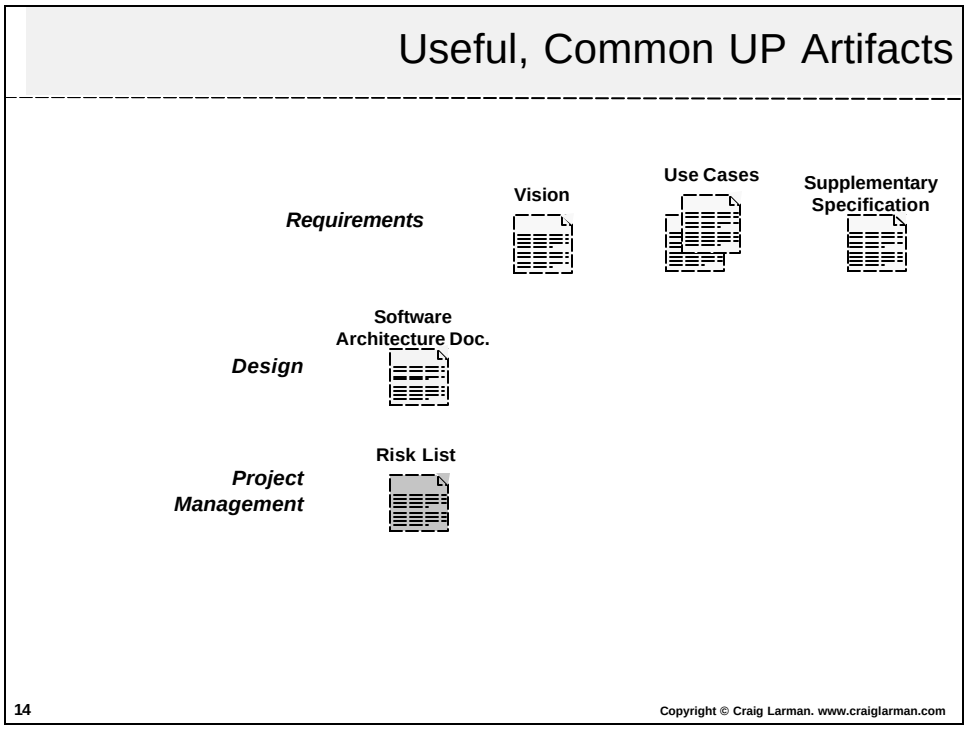
Discipline	Artifact	Incep.	Elab.	Const.	Trans.
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Project Management	SW Development Plan	s	r	r	r
Environment	Development Case	s	r		

12

Copyright © Craig Larman. www.craiglarman.com

Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com





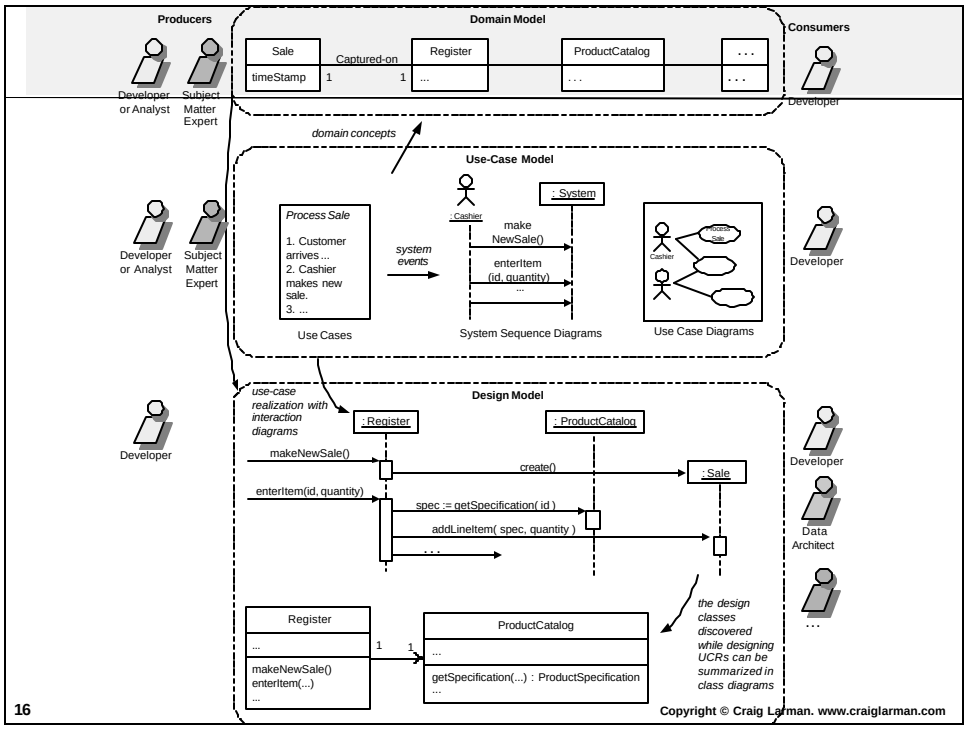
Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

When Work Products are Needed use the UP Vocabulary

- Vision
- Use Case Model
- Supplementary Specification
- Software Architecture Document

15

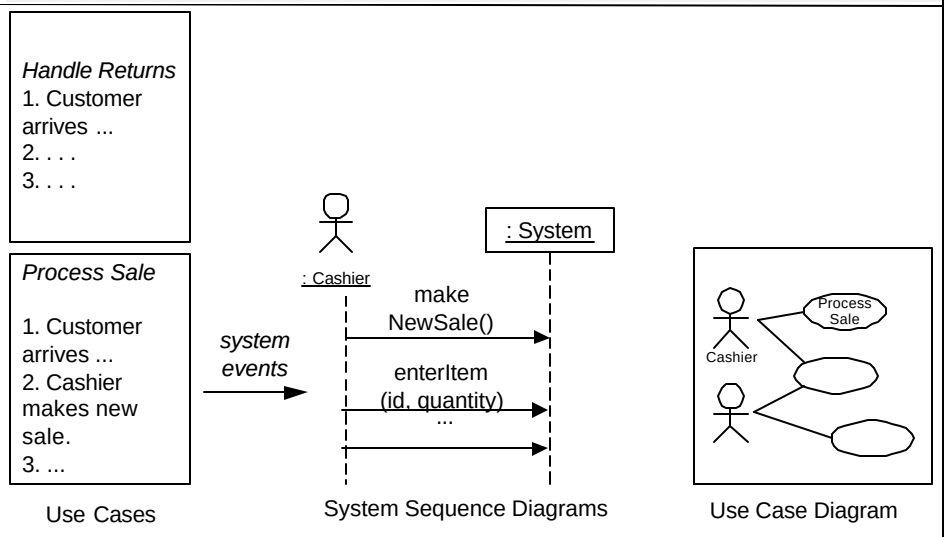
Copyright © Craig Larman. www.craiglarman.com



16

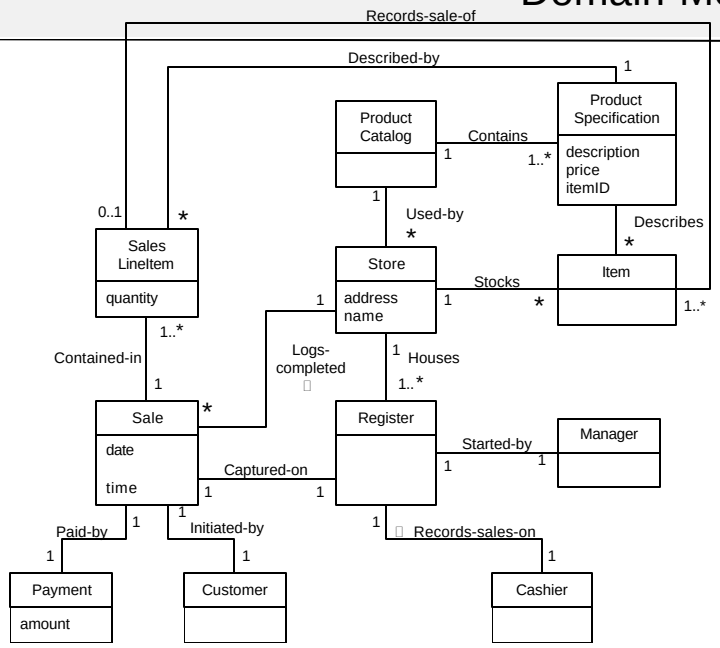
Copyright © Craig Larman. www.craiglarman.com

Use-Case Model



17 Copyright © Craig Larman. www.craiglarman.com

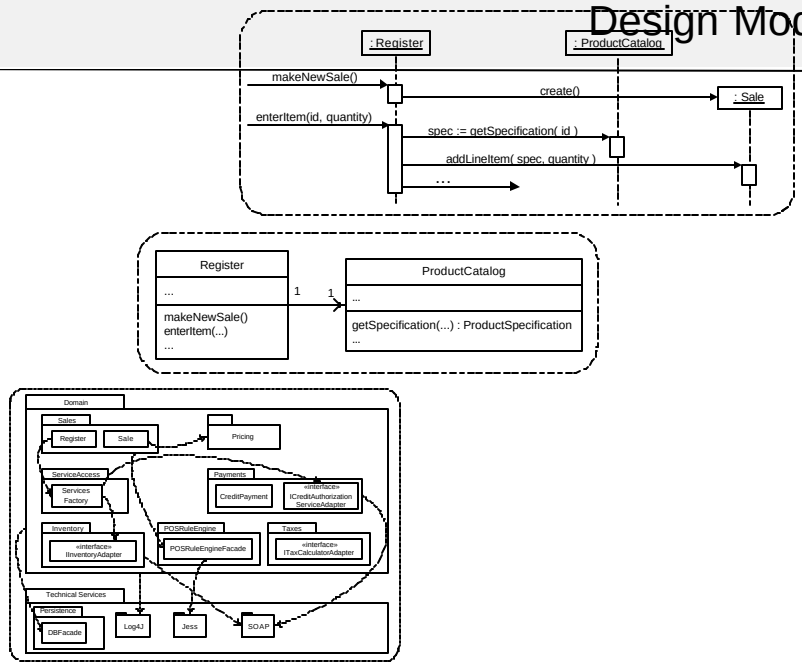
Domain Model



18 Copyright © Craig Larman. www.craiglarman.com

Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

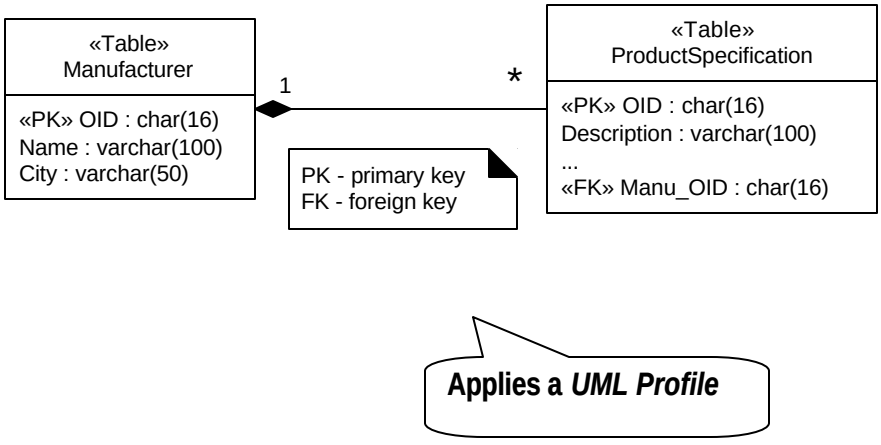
Design Model



19

Copyright © Craig Larman. www.craiglarman.com

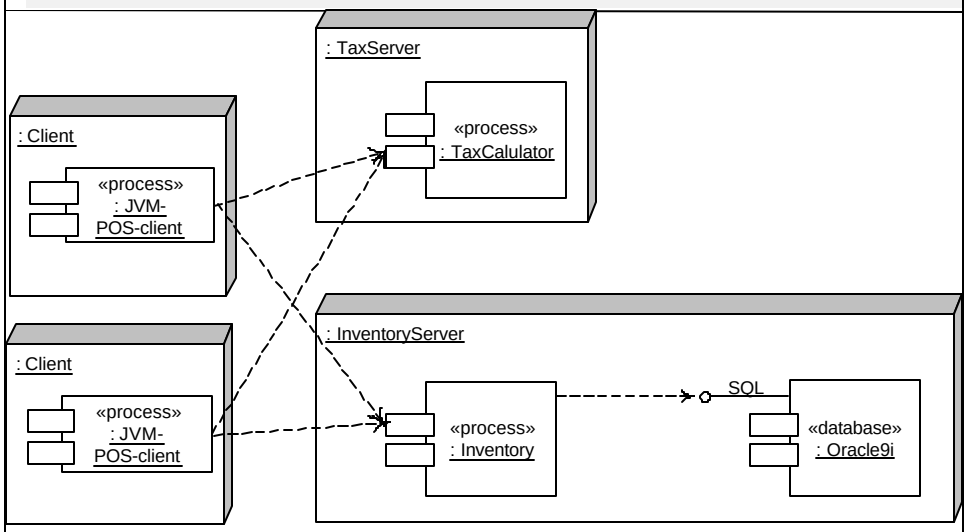
Data Model



20

Copyright © Craig Larman. www.craiglarman.com

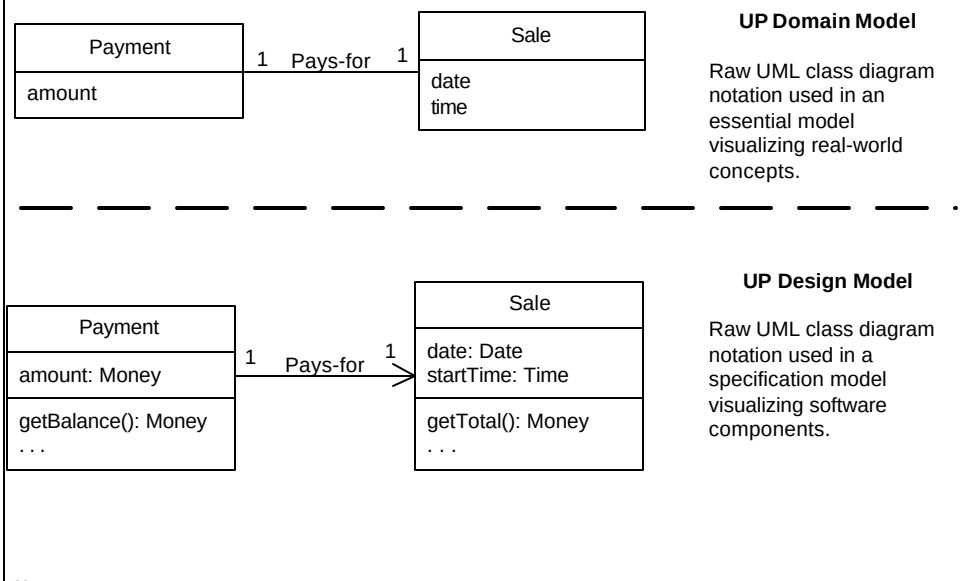
Deployment Model



21

Copyright © Craig Larman. www.craiglarman.com

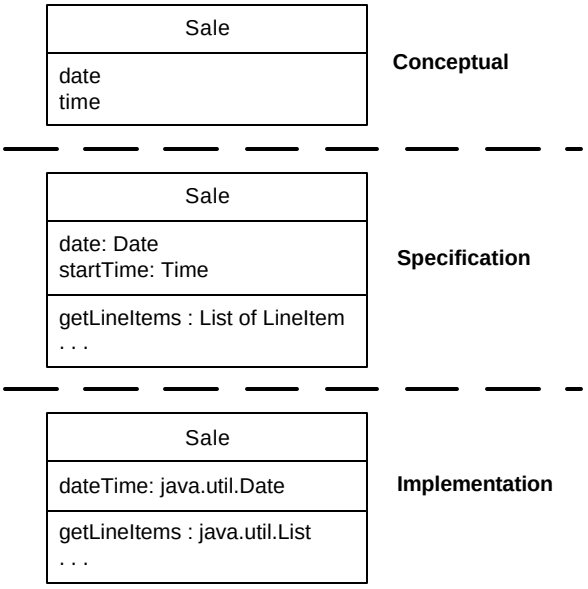
Notation vs. Method Models



22

Copyright © Craig Larman. www.craiglarman.com

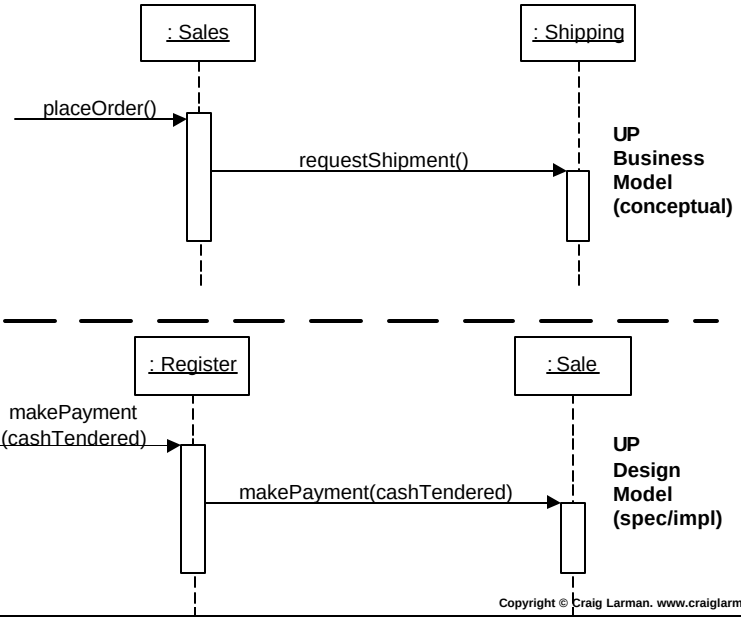
Three Modeling Perspectives (Cook & Daniels)



23

Copyright © Craig Larman. www.craiglarman.com

Modeling Perspectives in a Sequence Dgm

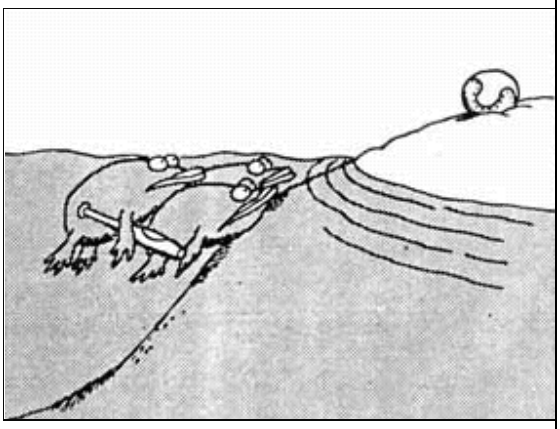


24

Copyright © Craig Larman. www.craiglarman.com

UP Adaptation

- In at least 4 areas:
  1. Requirements
  2. Design
  3. Plan
  4. Process

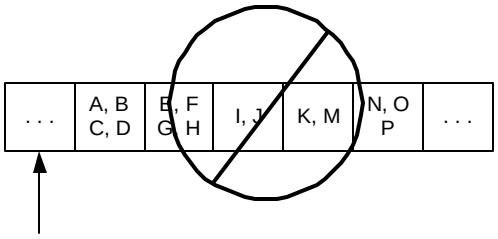


Great moments in evolution

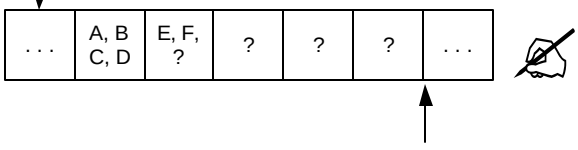
25

Copyright © Craig Larman. www.craiglarman.com

Adaptive Planning in the UP



Project is at this point (in the middle of an iteration)

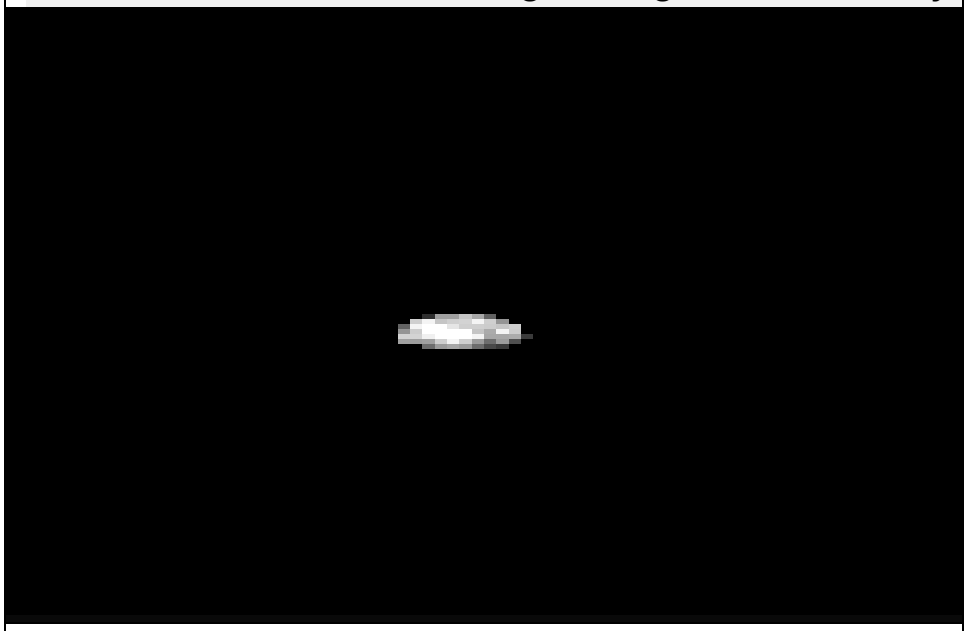


MILESTONE: A date (perhaps 2 months in future) and set of objectives is established.

26

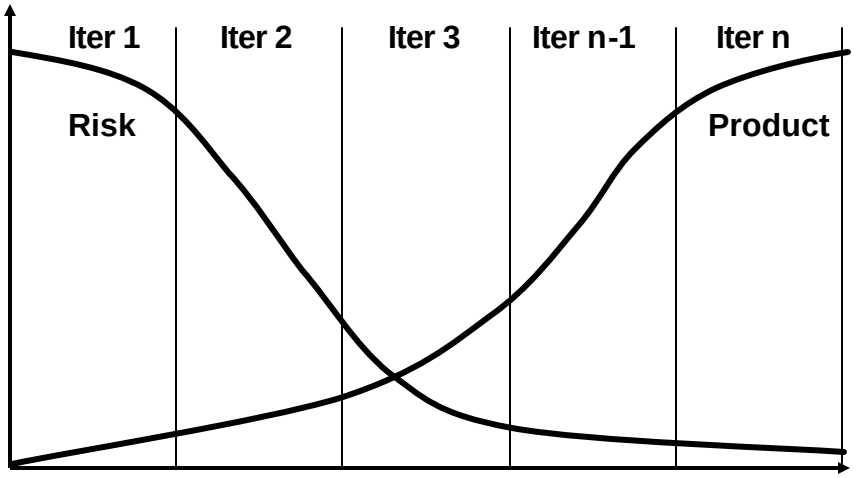
Copyright © Craig Larman. www.craiglarman.com

Mitigate High “Risks” Early



27 Copyright © Craig Larman. www.craiglarman.com

Mitigate High “Risks” Early



28 Copyright © Craig Larman. www.craiglarman.com

Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

Early Realistic Verification



Early evaluation —>



early feedback —>

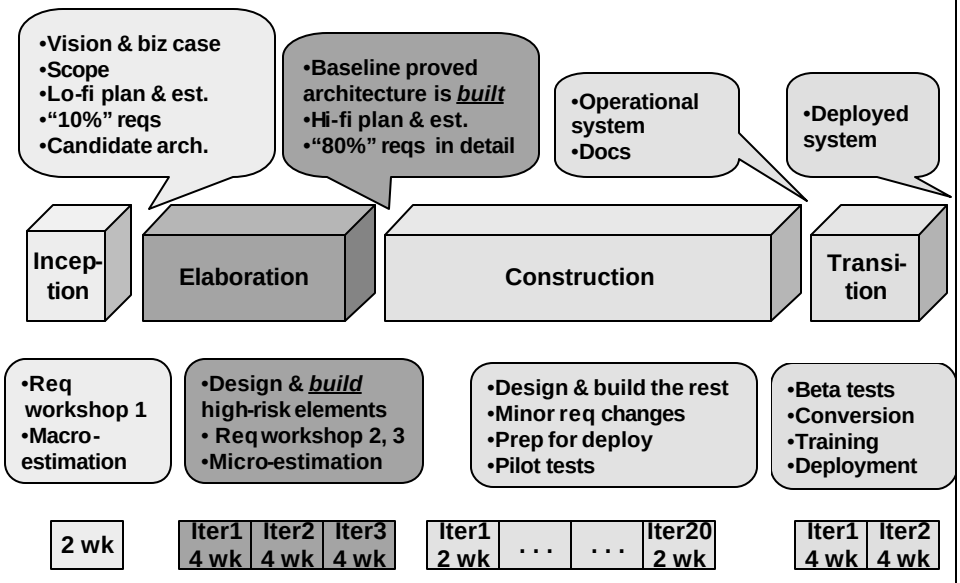


- **adaptation and better fit**
- **provable reliability**
- **early defect removal**

29

Copyright © Craig Larman. www.craiglarman.com

Follow the UP Phases



30

Copyright © Craig Larman. www.craiglarman.com

Follow the UP Phases

- The key distinctive idea is the *elaboration* phase.
  - Early iterations to discover and refine the core architecture and requirements while also *implementing* the production core of the system.
- Believable macro-level plan and estimates at the end of elaboration, not during inception.

31

Copyright © Craig Larman. www.craiglarman.com

Follow the UP Phases

Inception ? Requirements Phase

Elaboration ? Design Phase

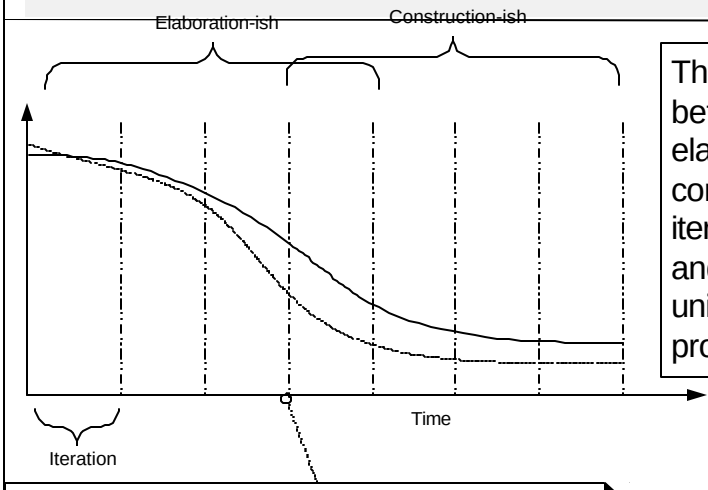
Construction ? Implementation Phase

Transition ? Integration/QA Phase

32

Copyright © Craig Larman. www.craiglarman.com

Elaboration vs. Construction?



The distinction between elaboration and construction iterations is fuzzy, and often unimportant on a project.

On some projects, a milestone decision to stop or to submit the bidding for out-sourced development.

risk

requirements instability

33

Copyright © Craig Larman. www.craiglarman.com

Elaboration vs. Construction?

- Therefore, ask yourself: “Is there some real practical reason I need to distinguish between elaboration and construction?”
- Often, the answer is no. Fine.
- Then we have:
  1. Inception—a few days or weeks to vaguely comprehend the project goals and scope.
  2. “Development”—a series of iterations, driven by risk, that continue until transition.
  3. Transition—beta test and deploy.

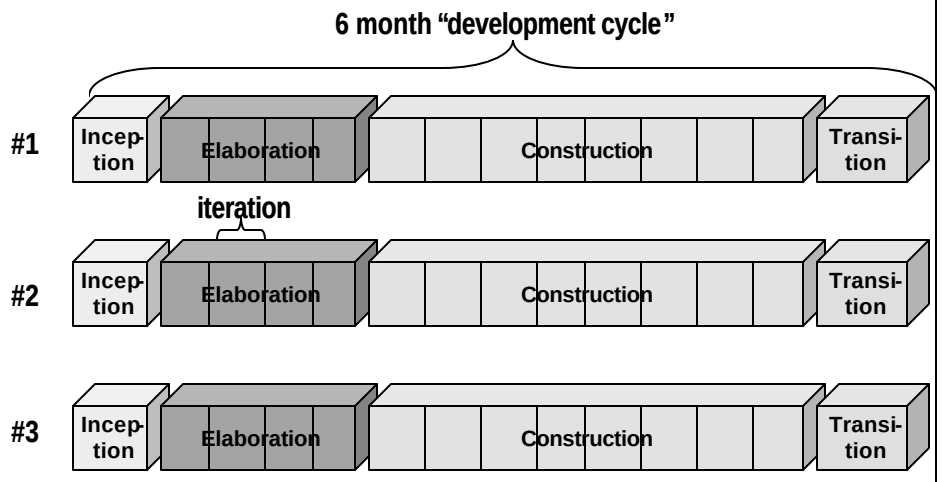
34

Copyright © Craig Larman. www.craiglarman.com

Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

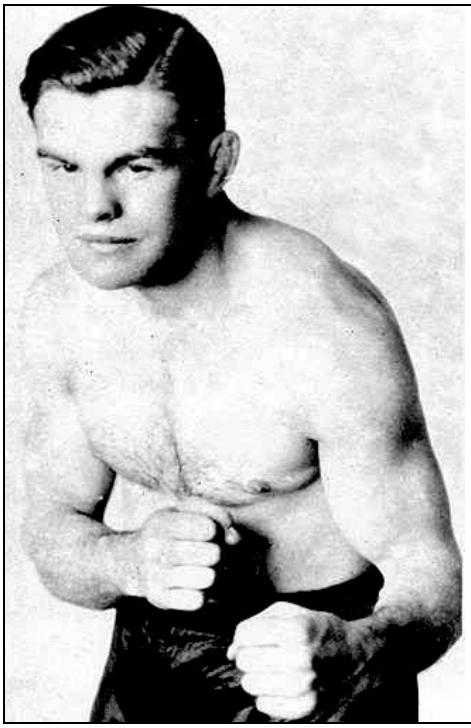
Where Possible, Incremental Delivery

- Partial system in production, early.
- Difficult with embedded or device systems; easy with web sites.



35

Copyright © Craig Larman. www.craiglarman.com



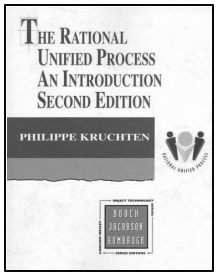
Why the UP?

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

History and Rationale

- The UP has rapidly emerged as a world-wide de facto industry standard for modern process.
- Groups within many leading companies are adopting it.
  - IBM, Chase-Manhattan, Telia, Valtech, Alcatel, MCI, British Aerospace, Sprint, Volvo, Intel, Merrill, E&Y, Deloitte, Ericsson, . . .



37

Copyright © Craig Larman. www.craiglarman.com

History and Rationale

Why acceptance?

- It is good—combines most popular modern best practices (e.g., iterative development)
- Perceived as coming from modern process experts.
- It is cohesive and well documented
  - HTML based RUP product
  - Books

38

Copyright © Craig Larman. www.craiglarman.com

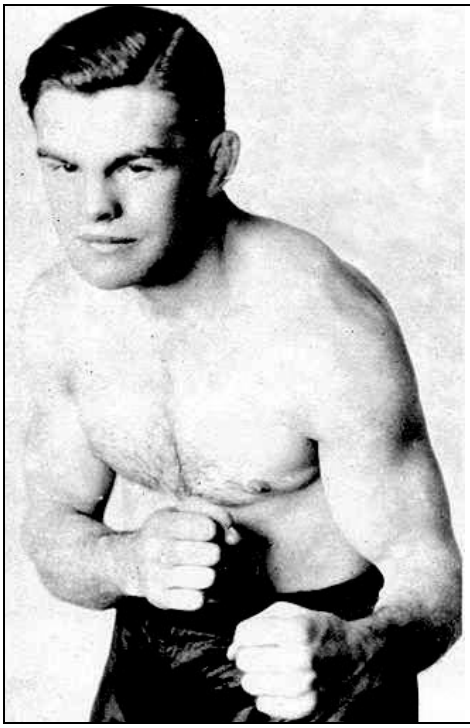
History and Rationale

Why acceptance?

- Flexible.
- Many organizations have “process fatigue” and just want to adopt a good “turn key” solution.

39

Copyright © Craig Larman. www.craiglarman.com



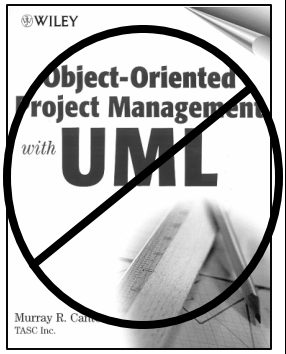
How to Fail with the UP

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

Warning: Superimposing Waterfall Thinking

- There are books that give the impression of being a source of UP advice.
- They are based on this (incorrect) viewpoint:
  - Inception—do the requirements
  - Elaboration—do the design or models
  - Construction—translate design to code
- Warning! People superimpose waterfall ideas/values on the UP.



41

Copyright © Craig Larman. www.craiglarman.com

Kruchten on Waterfall versus Iterative

"In a waterfall approach, there is a lot of emphasis on "the specs" and getting them right, complete, polished and signed-off. In the iterative process, the *software* we develop comes first. It is the software product that is the main focus of attention throughout, with both specs and software evolving in parallel. This has impact on the teams: testers, for example, may be used to receiving complete, stable specs, with plenty of advanced notice to start testing; whereas in an iterative development, they have to begin working at once, with specs and requirements that are still evolving."

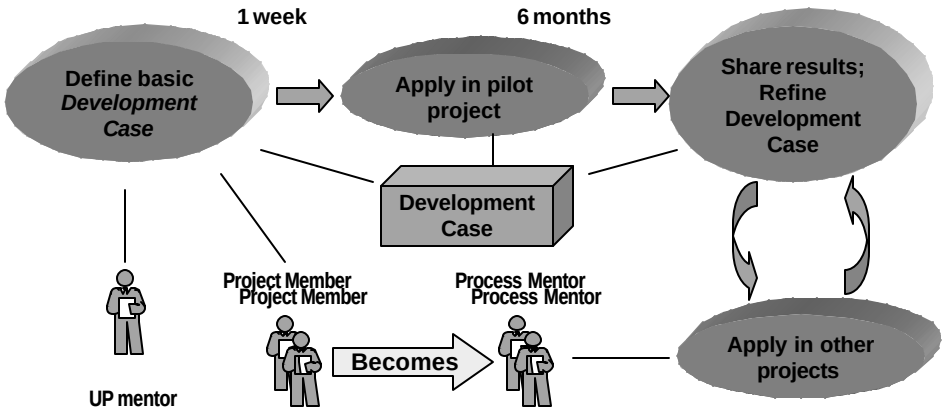
From Waterfall to Iterative Lifecycle – A tough transition for project managers

by Philippe Kruchten

42

Copyright © Craig Larman. www.craiglarman.com

Adopt/Build a Process like a System—Iteratively and Incrementally



43

Copyright © Craig Larman. www.craiglarman.com

How to Fail with the UP

- No one has experience with short timeboxed iterations.
- There is not a critical mass of very knowledgeable object-oriented developers.
- Management doesn't really understand the profound implications of moving from waterfall to iterative lifecycle and adaptive attitudes.

44

Copyright © Craig Larman. www.craiglarman.com

Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

How to Fail with the UP

- Big bang process adoption
- Leaders implementing iterative development and the UP don't have direct experience working as developers in iterative development and object technology projects.
- Defining changes or "enhancements" to the UP, when have not yet practiced and mastered the basics.

45 Copyright © Craig Larman. www.craiglarman.com

How to Fail with the UP

- First iteration too big or ambitious.
- Not intensively testing and integrating in each iteration.
- Not doing true iterative development—still applying waterfall ideas.
  - Front-loading . . .

46 Copyright © Craig Larman. www.craiglarman.com

How to Fail with the UP

- Creating too many artifacts.
- Focusing on learning or doing UP minutiae rather than the grasping the big ideas.

47 Copyright © Craig Larman. www.craiglarman.com

You know you didn't understand the UP when…

- Inception = requirements,
  - elaboration = design, . . .
- One tries to define most of the requirements before starting design or implementation.
- One tries to define most of the design before starting implementation.

48 Copyright © Craig Larman. www.craiglarman.com

You know you didn't understand the UP when…

- "Iterations" in months instead of weeks.
- One tries to speculatively predict all iterations, and what will happen in each one.
- An organization wants believable plans and estimates for projects before they have entered the elaboration phase.

49

Copyright © Craig Larman. www.craiglarman.com