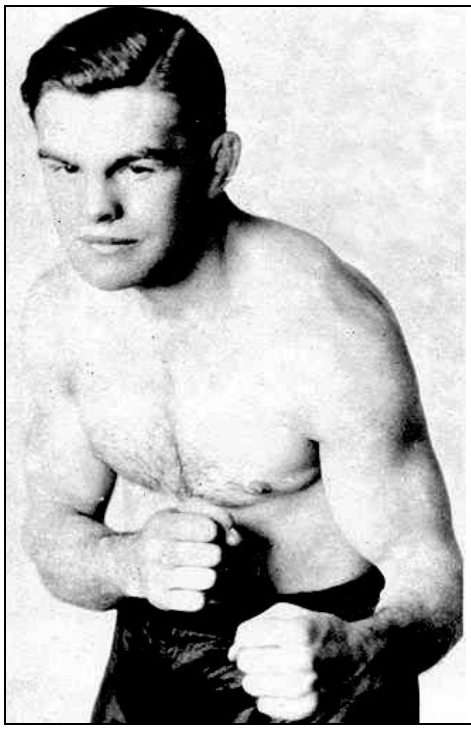
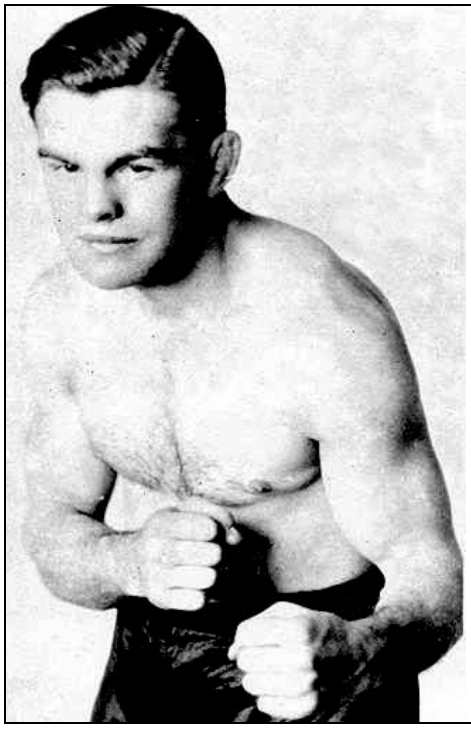


An Agile UP: Details and XP

Craig Larman

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.



UP & XP

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

XP Sure Looks Heavy

So many rules must be followed!

- Pair programming
- Test-first programming
- Continuous integration
- Common project room
- On-site customer
- Planning game
- Simple design
- Collective code ownership
- 40 hour week
- Coding standards
- Refactoring
- Metaphor

So many artifacts to create! **(quoted from the XP books)**

- Story cards
- Feature task cards
- Engineering task cards
- CRC cards
- Visible graphs
- Big Plan
- Story estimates
- Task estimates
- Unit tests
- Acceptance tests
- Grouped defects to fix story cards
- Problem/bug cards
- Release Plan
- Velocity measurements
- Coding standards
- Progress report
- Unit test score report
- Acceptance test score report
- Time spent on defects report
- Code

3

Copyright © Craig Larman. www.craiglarman.com

UP Sure Looks Light

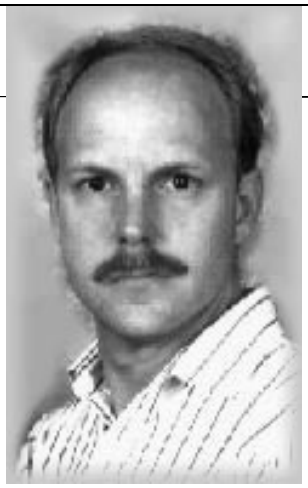
No strict rules.

Only one artifact is required.

Code

4

Copyright © Craig Larman. www.craiglarman.com









5

.craiglarman.com

Loss of... Life	L6	L20	L40	. . .
Essential $$	E6	E20	E40	. . .
Discre-tionary $$	D6	D20	D40	. . .
Comfort	C6	C20	D40	. . .
	1-6 people	-20	-40	. . .

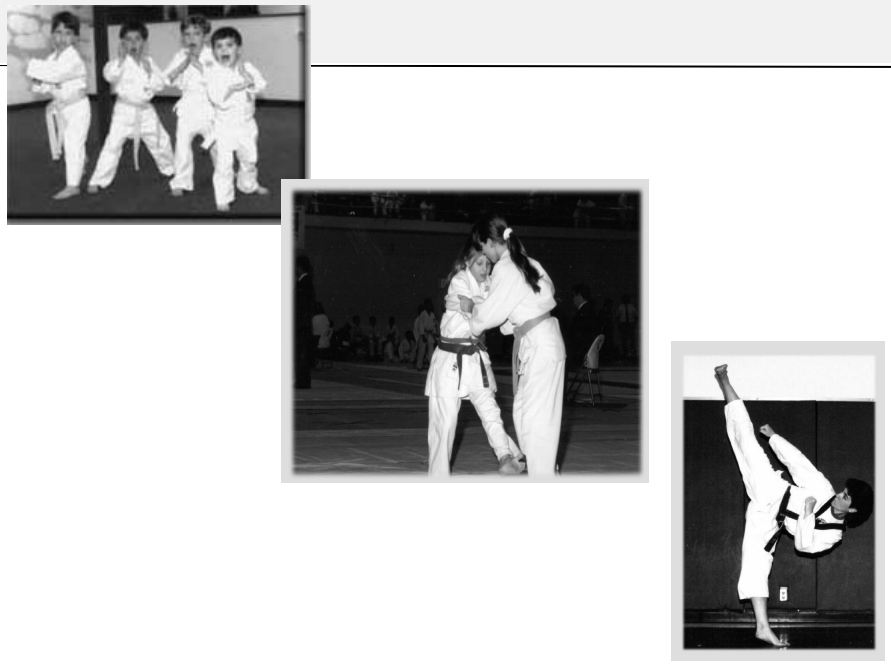
UP

XP



6

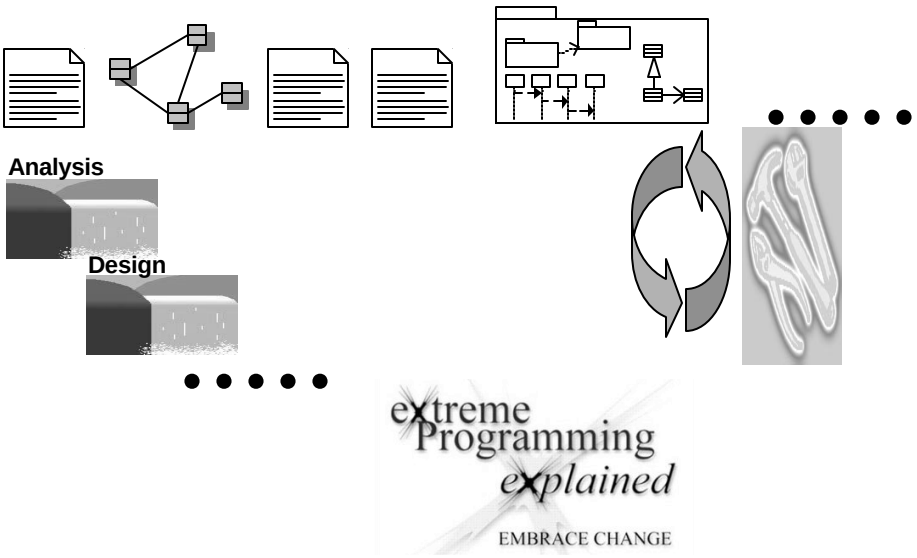
Copyright © Craig Larman. www.craigl



7

Copyright © Craig Larman. www.craiglarman.com

Myths About UP



8

Copyright © Craig Larman. www.craiglarman.com

Myths About XP

Pair Programming

Test First Programming

Onsite Customer

Continuous Integration

• • •





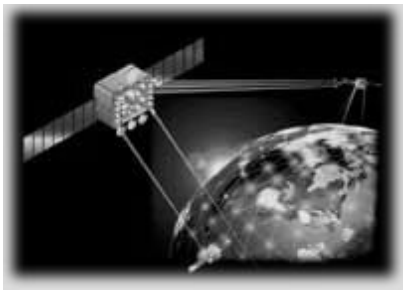


9

Copyright © Craig Larman. www.craiglarman.com

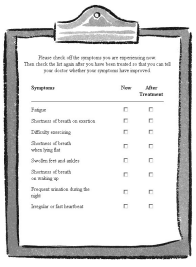
Why Need More than XP?









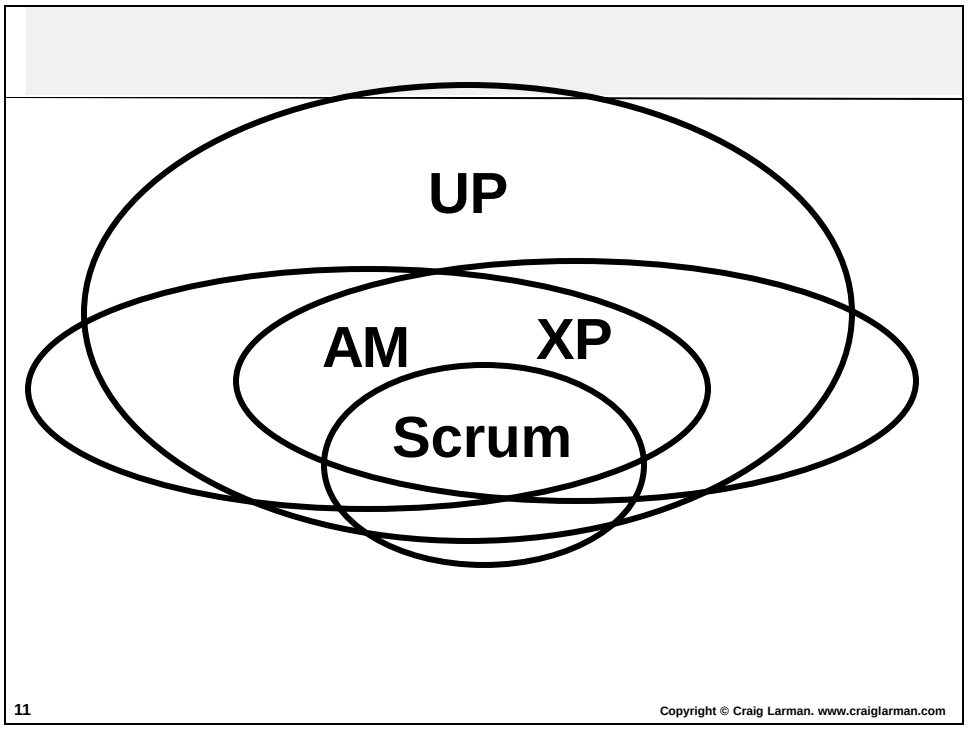


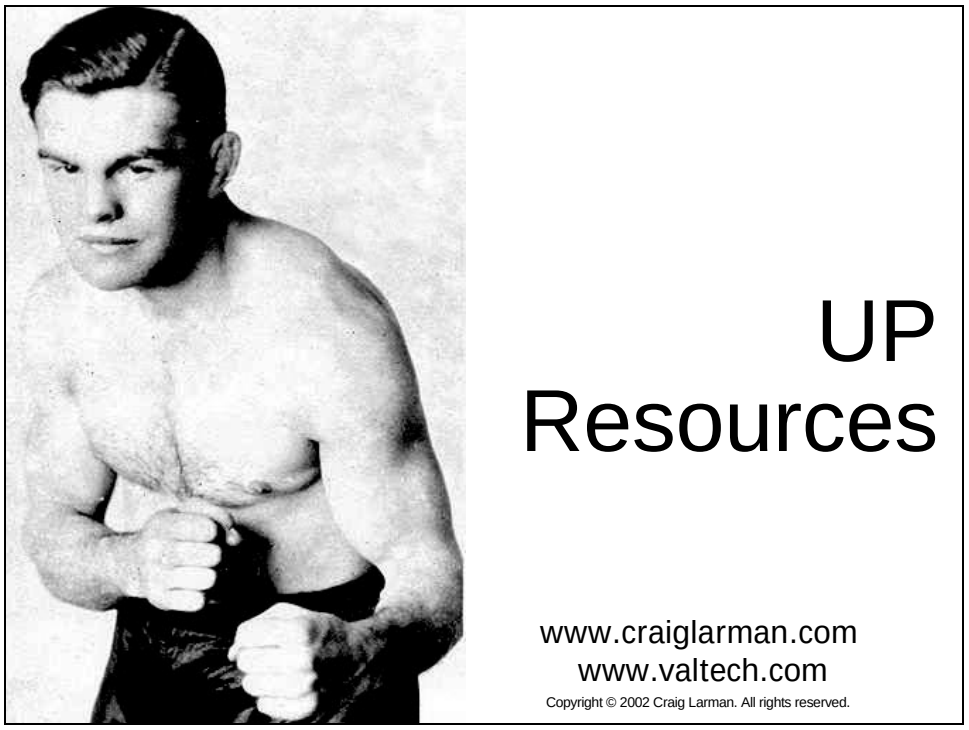
Please check off the symptoms you are experiencing now.
Then check the list again after you have been treated so that you can tell your doctor whether your symptoms have improved.

Symptoms	Now	After Treatment
Fatigue	☐	☐
Shortness of breath on exertion	☐	☐
Difficulty exercising	☐	☐
Shortness of breath when lying flat	☐	☐
Swollen feet and ankles	☐	☐
Shortness of breath on waking up	☐	☐
Frequent urination during the night	☐	☐
Irregular or fast heartbeat	☐	☐

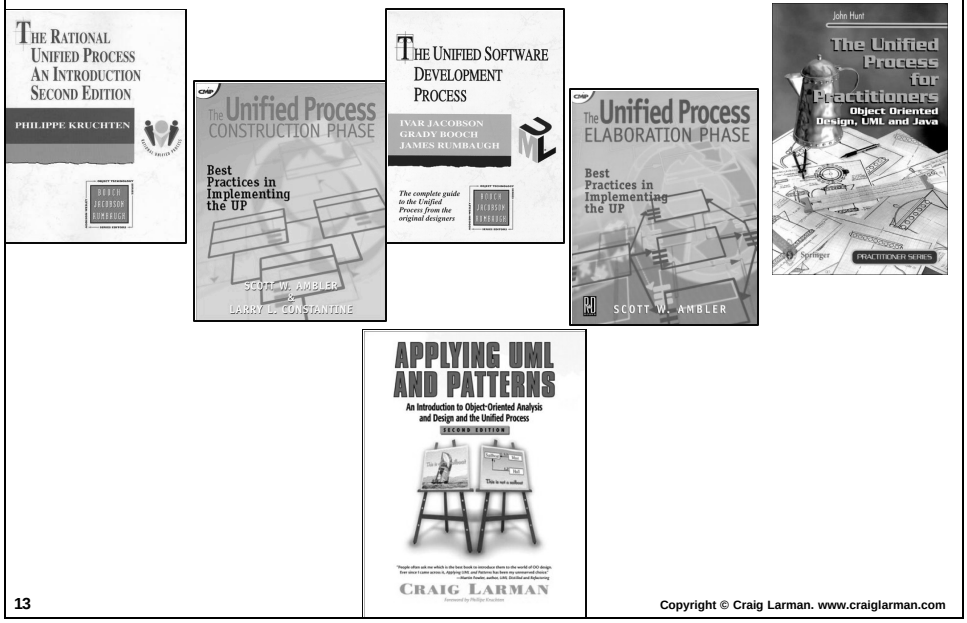
10

Copyright © Craig Larman. www.craiglarman.com


UP
AM
XP
Scrum
11
Copyright © Craig Larman. www.craiglarman.com


UP Resources
www.craiglarman.com
www.valtech.com
Copyright © 2002 Craig Larman. All rights reserved.

Sample UP Resources



13

Copyright © Craig Larman. www.craiglarman.com

Sample UP Resources

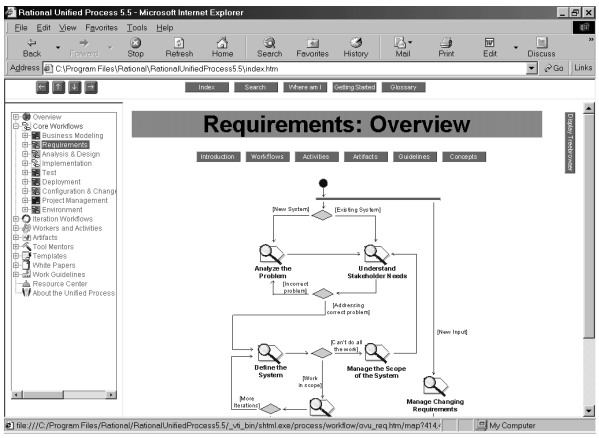
- www.craiglarman.com
- www.rational.com
- www.ambysoft.com
- www.agilemodeling.com

14

Copyright © Craig Larman. www.craiglarman.com

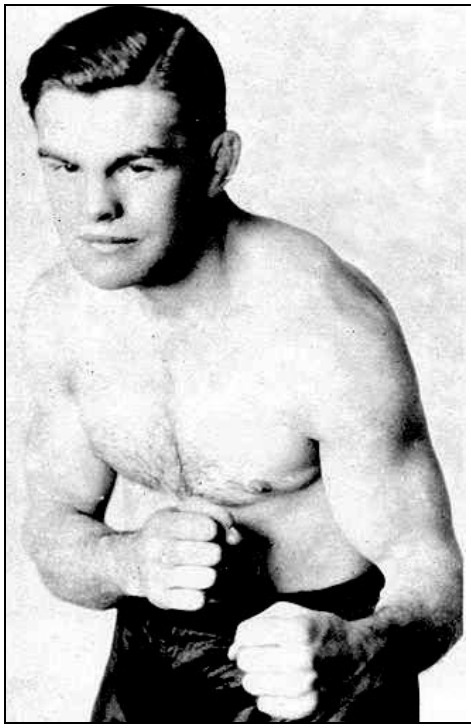
The RUP Product

- Several thousand pages of definition, guidance, templates.
- Since it is a set of HTML pages, may be placed on an intranet (licensing respected).



15

Copyright © Craig Larman. www.craiglarman.com

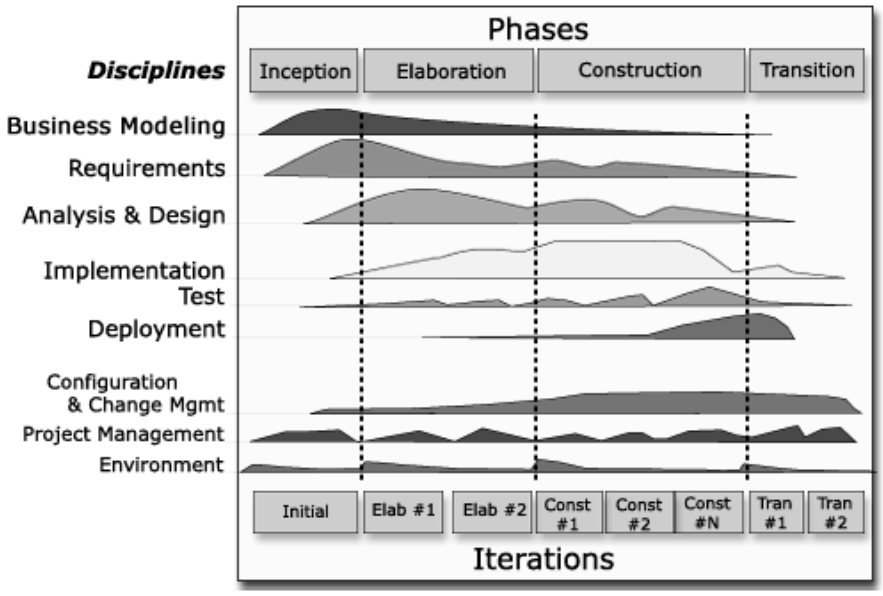


More Core UP Ideas

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

UP Disciplines

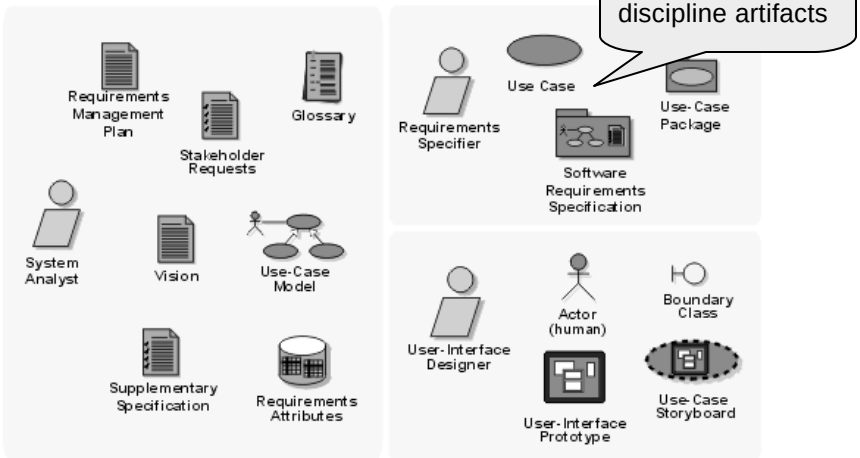


17

Copyright © Craig Larman. www.craiglarman.com

Disciplines and Artifacts

- Artifacts exist within disciplines.
- All artifacts are optional.



18

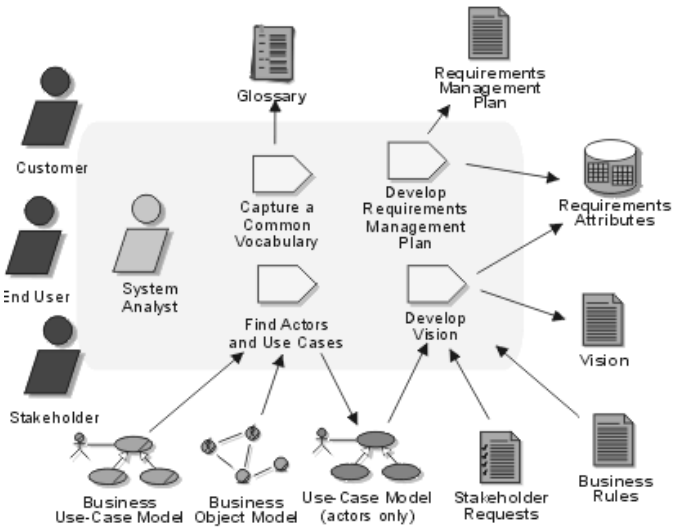
Copyright © Craig Larman. www.craiglarman.com

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

Disciplines, Artifacts, and Activities

- Artifacts are created within disciplines during an activity.

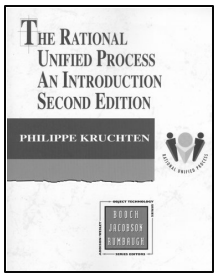
Some activities within the requirements discipline



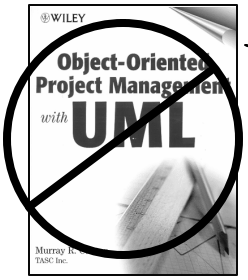
19

Copyright © Craig Larman. www.craiglarman.com

What the Phases are Not. Warning: Confusing Book Descriptions



Warning:
The long sections accurately describe the phases, but the short bullet-point summaries can give a misleading "waterfall" impression.

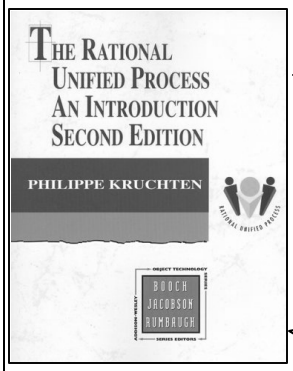


Avoid.
Just plain wrong.

20

Copyright © Craig Larman. www.craiglarman.com

What the Phases are Not. Warning: Confusing Book Descriptions



Elaboration Summary (p. 62)

"…specifying the features and designing the architecture."

Without further reading, can be misunderstood as a "waterfall-ish" requirements & design phase that doesn't implement and test the core architecture, and discover based on an iterative approach.

Elaboration Details (p. 69, 70)

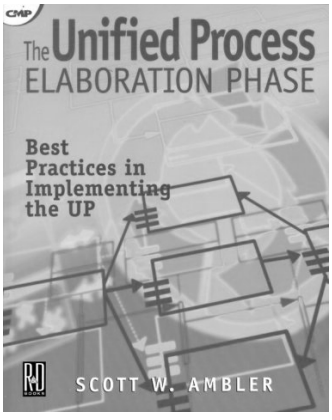
"In the Elaboration phase, an executable architecture is built in one or more iterations".

"Define, validate, and baseline the architecture as rapidly as possible."

OK, that's a better picture

21 Copyright © Craig Larman. www.craiglarman.com

What the Phases are Not. Warning: Confusing Book Descriptions



Incorrectly describes the development work as throw-away prototype work.

The code for the architectural baseline should be *production code*.

However, there is of course room for *some* throw-away prototypes just implemented to explore and prove out some risky issue.

22 Copyright © Craig Larman. www.craiglarman.com

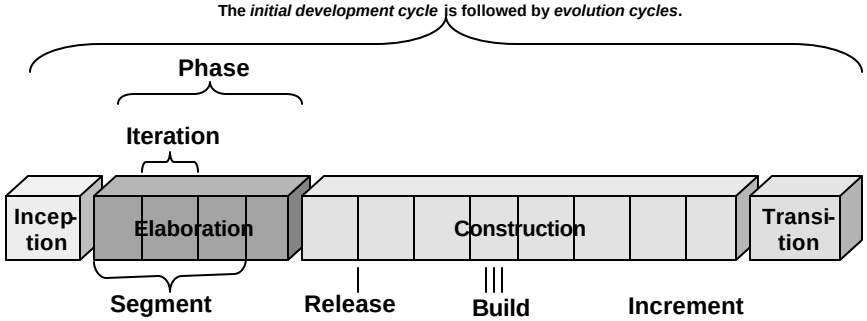
Copyright (c) 2002 Craig Larman. All rights reserved. www.craiglarman.com

Important UP Terminology—Schedule Oriented

Development Cycle

Ends in a *final production release*.

The *initial development cycle* is followed by *evolution cycles*.



Segment

Valtech term only!

A group of iterations for planning or milestone purposes. For example, for a pilot test milestone, or to end with a short iteration focusing just on performance.

Release

A stable executable subset of the final product. The end of each iteration is a minor release.

Build

An operational version of the system or a portion. e.g., "the daily build."

Increment

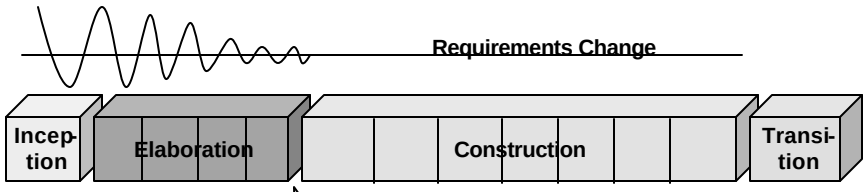
The difference (delta) between the releases of 2 subsequent iterations.

23

Copyright © Craig Larman. www.craiglarman.com

Requirements Always Unstable?

- No, requirement instability does exists in Elaboration, but most requirements are stabilized by the end of Elaboration.



During Elaboration, we iteratively discover, clarify, and refine most of the requirements, while also building the core architecture.

"80"% of the requirements clarified and stabilized

The Construction phase is more of a stable manufacturing phase with relatively clear and stable requirements

"20%" change.

24

Copyright © Craig Larman. www.craiglarman.com

Length of Iterations

- "Ideally, an iteration should span from two to six weeks."
- 1 -30 people : 2-4 weeks
- 30 - 50 : 2-6 weeks
- Factors that increase length:
  - More people
  - More discovery/creativity
  - Geographically dispersed
  - Team new to the technologies or process

25 Copyright © Craig Larman. www.craiglarman.com

Time-boxing and Iterations

- Time-boxed: End date is fixed; doesn't vary.
  - If can't meet goals, scale back requirements for this iteration
- Time-boxing is a key concept in iterative development.
- "Last day" may include a demo to external stakeholders

26 Copyright © Craig Larman. www.craiglarman.com

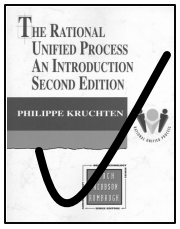
Time-boxing and Iterations

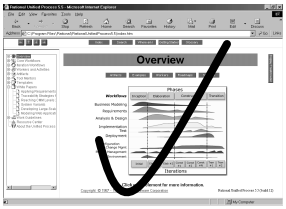
Advantages:

- Forces the team to focus on and prioritize goals
- Counter-balance to “work expands to fill the time allotted”.
- Focus and energy generated by a fixed, soon-coming date.
- Completing these formal mini-milestones increase feelings of closure, resolution, satisfaction, competency.

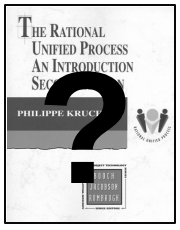
27 Copyright © Craig Larman. www.craiglarman.com

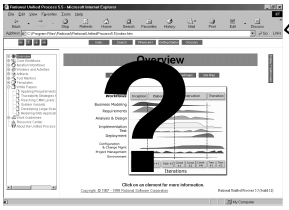
What about the RUP Product/Book Advice?





These are excellent for advice on the process steps and elements.





Not for (serious) advice on skillfully doing a particular step (e.g., object design, architectural analysis).

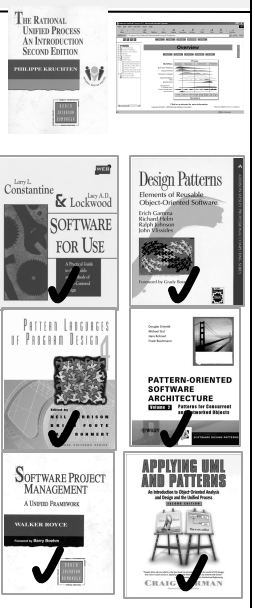
The RUP material just hints at the real knowledge needed.

Not Rational’s fault—too vast topics.

28 Copyright © Craig Larman. www.craiglarman.com

What about the RUP Tool/Book Advice?

- The documents are good at *breadth*, not depth.



. . .

29

Copyright © Craig Larman. www.craiglarman.com

Where is the UML Used?

- For example. . .



Design Model



Package diagrams

Static structure (class) diagrams

Interaction diagrams

Most common

Use-Case Model



Use case diagrams

Deployment Model

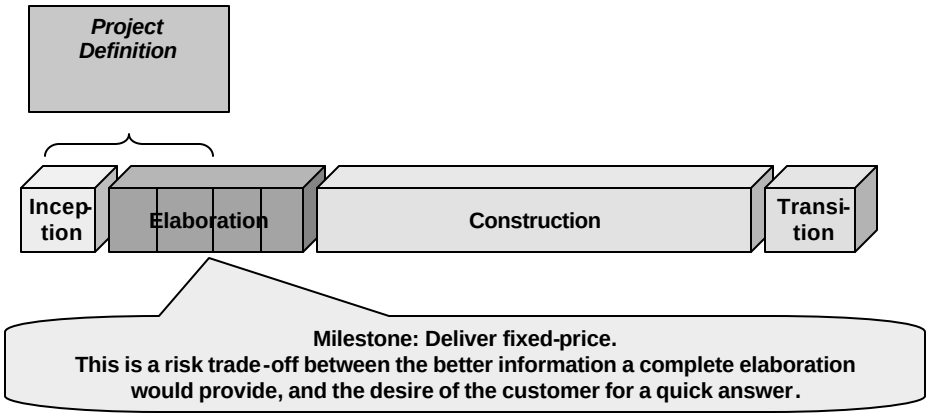


Deployment diagrams

30

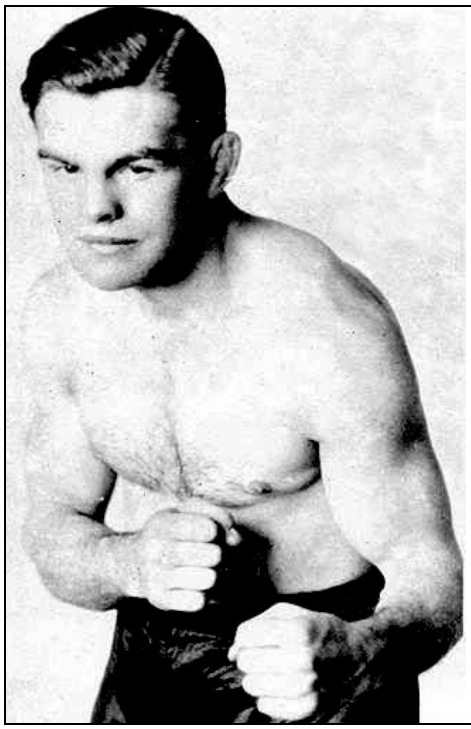
Copyright © Craig Larman. www.craiglarman.com

Fixed-Price Contracts



31

Copyright © Craig Larman. www.craiglarman.com

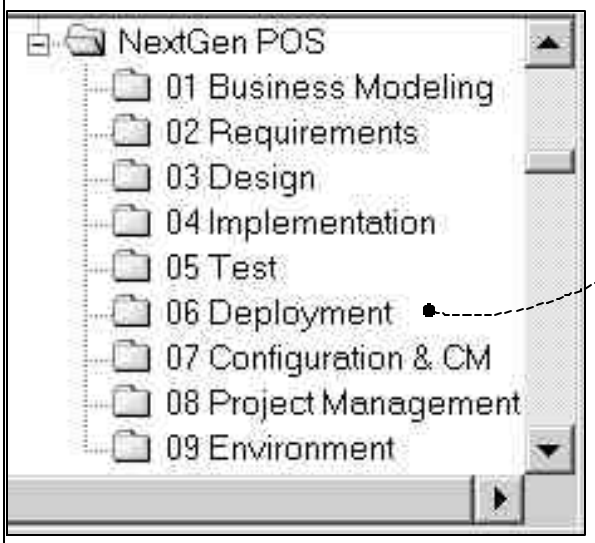


Discipline and Artifact Comments

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

Disciplines and Folder Structure



33

Copyright © Craig Larman. www.craiglarman.com

Core Artifacts?

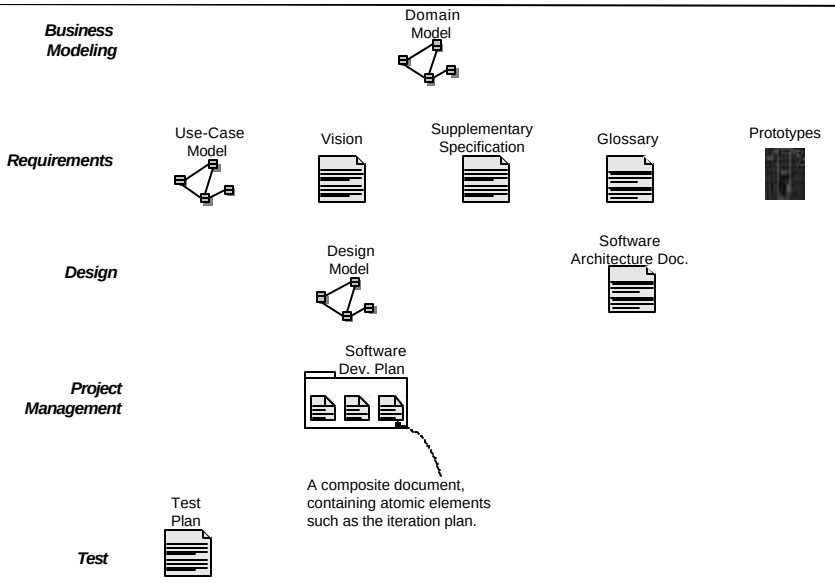
- What are common UP artifacts found on "*many*" projects?



34

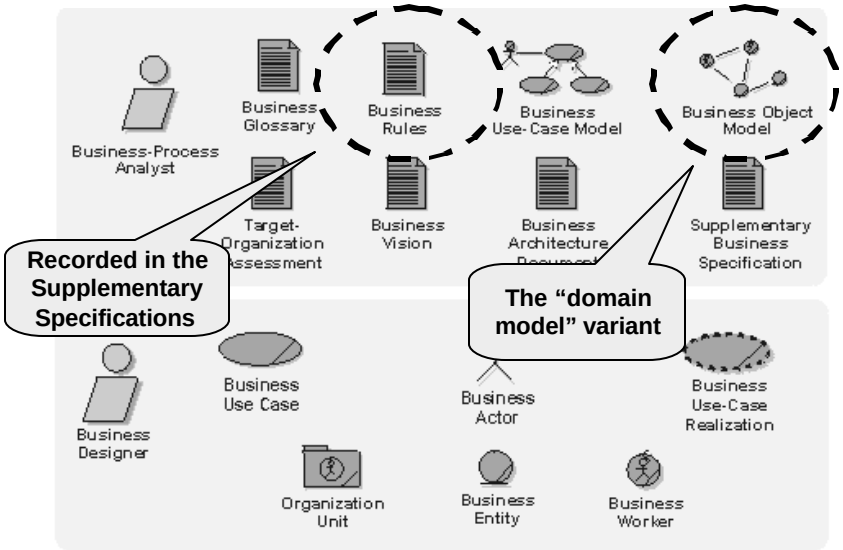
Copyright © Craig Larman. www.craiglarman.com

A Simple Development Case



35 Copyright © Craig Larman. www.craiglarman.com

Business Modeling Artifacts

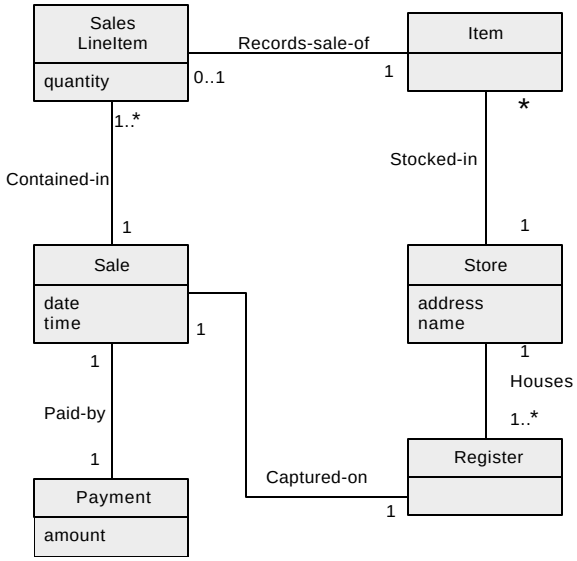


36 Copyright © Craig Larman. www.craiglarman.com

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

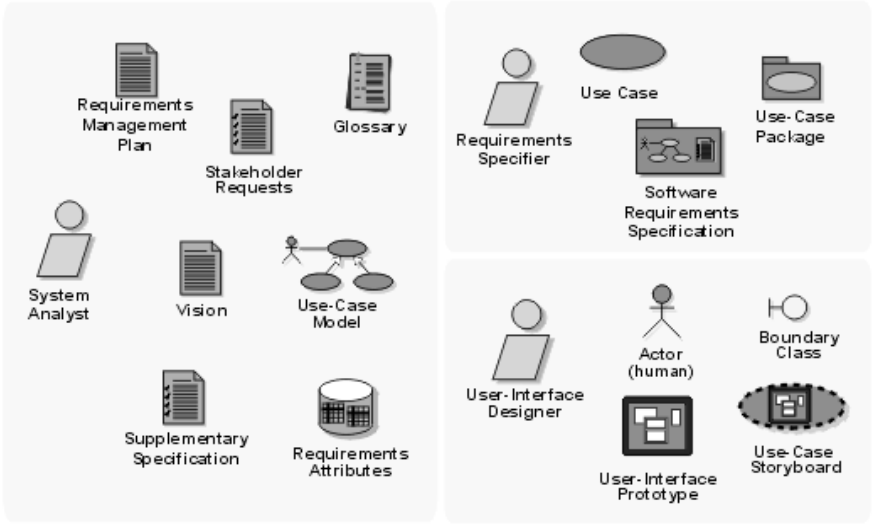
Business Modeling—Domain Model

- A visual dictionary of noteworthy concepts.
- Not a picture of software components.



37 Copyright © Craig Larman. www.craiglarman.com

Requirements Artifacts



38 Copyright © Craig Larman. www.craiglarman.com

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

Use-Case Model

In the UP use cases are the central requirements idea; use cases should “permeate” requirements work and thinking.

- If on a project use cases feel like “just another” addition to the requirements set, something is wrong.
  - They should be the focus and main effort, connecting all requirements.

39 Copyright © Craig Larman. www.craiglarman.com

Use Cases are Text Documents

- Use cases are written **text** documents, not diagrams.
- The Use Case Model is really the *written* text use cases.
- Use case diagrams and relationships (*includes*, *extends*, …) are not very important.

40 Copyright © Craig Larman. www.craiglarman.com

Supplementary Specification

- Non-functional and all other requirements:
  - usability, reliability, performance, . . .
  - reports
  - hardware and software constraints (operating and networking systems, ...)
  - development constraints (for example, process or development tools)
  - internationalization concerns . . .
  - documentation
  - . . .

41 Copyright © Craig Larman. www.craiglarman.com

Requirements Attributes

- The attribute-value pairs attached to each requirement, such as:
  - use case
  - nonfunctional requirement
  - feature
- Common attributes:
  - priority
  - status (requested, approved, underway, …)
  - risk, . . .
- Best managed with a requirements tool.
- Critical to the UP best practice of “requirements management”

42 Copyright © Craig Larman. www.craiglarman.com

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

Other Frequent Artifacts



Vision



Glossary

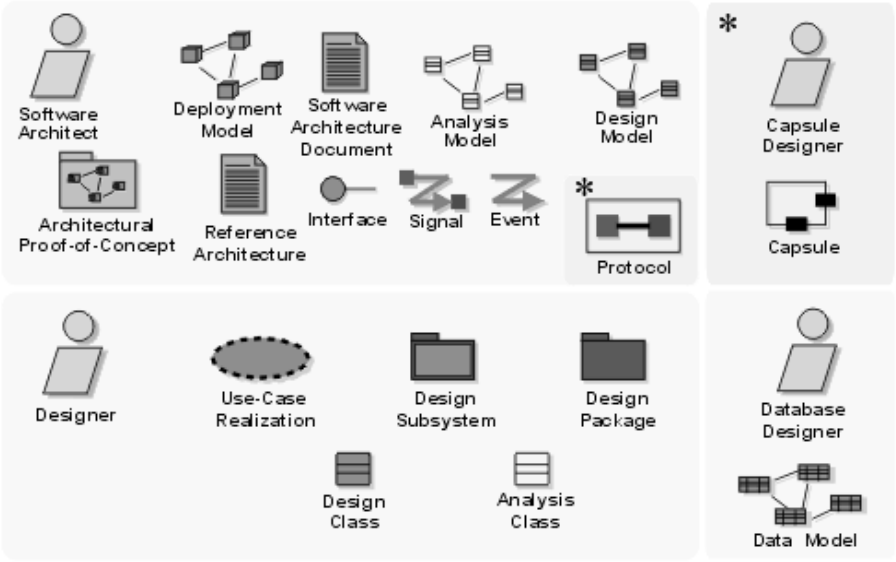


UI Prototypes and specs

43

Copyright © Craig Larman. www.craiglarman.com

Design Artifacts



44

Copyright © Craig Larman. www.craiglarman.com

The Design Model

- The object design of the system.
  - Packages, frameworks, classes, interfaces, …
- "use case realizations"
- The model in which visual modeling is used most frequently.
  - Package diagrams
  - Interaction diagrams
  - Class diagrams

45 Copyright © Craig Larman. www.craiglarman.com

The Software Architecture Document and Views

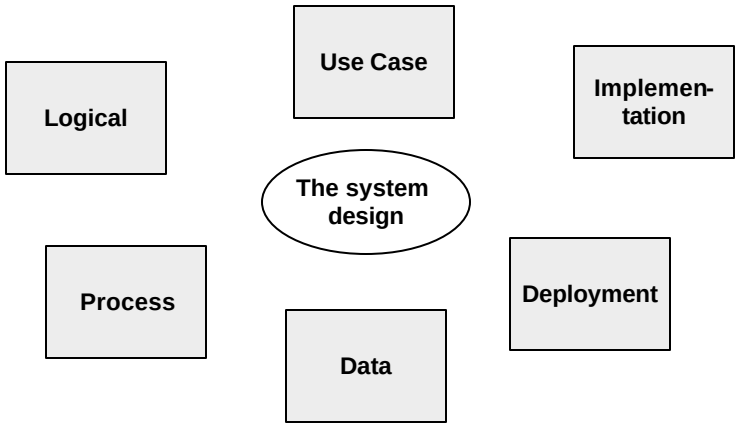
- What if you want to communicate a summary to someone?
  - When joining the project, when modifying it, …
- In the UP, architectural views are used.
- The critical 1-minute "elevator descriptions."



46 Copyright © Craig Larman. www.craiglarman.com

Architectural Views

- At present, the UP defines “N” views
  - this replaces the old 4+1 view approach



47

Copyright © Craig Larman. www.craiglarman.com

SAD – Motivation and Design Diary

- Many architecture definitions unfortunately just emphasize describing the structure.
- What’s missing?

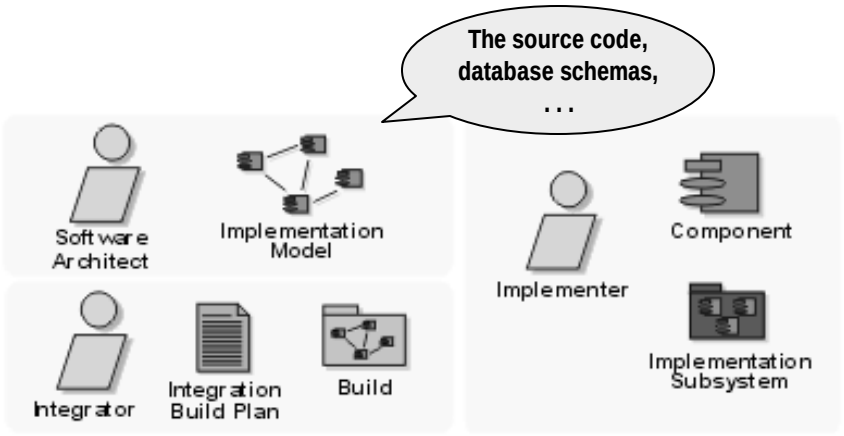
Motivation

- Recommended practice:
  - Maintain a design diary, and record the motivation for the design decisions.
    - Insert this into the Software Architecture Document.

48

Copyright © Craig Larman. www.craiglarman.com

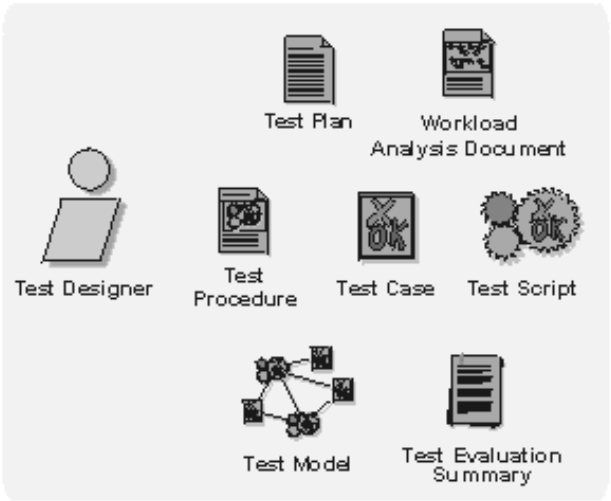
Implementation Artifacts



49

Copyright © Craig Larman. www.craiglarman.com

Testing Artifacts



50

Copyright © Craig Larman. www.craiglarman.com

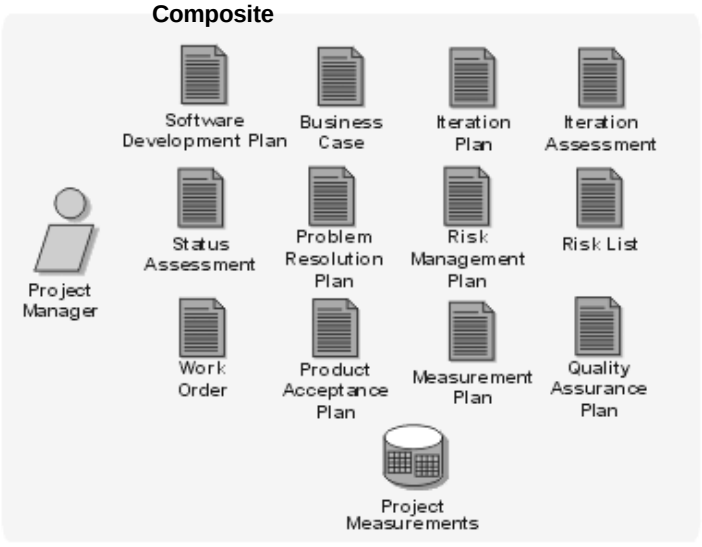
Deployment Artifacts



51

Copyright © Craig Larman. www.craiglarman.com

Project Management

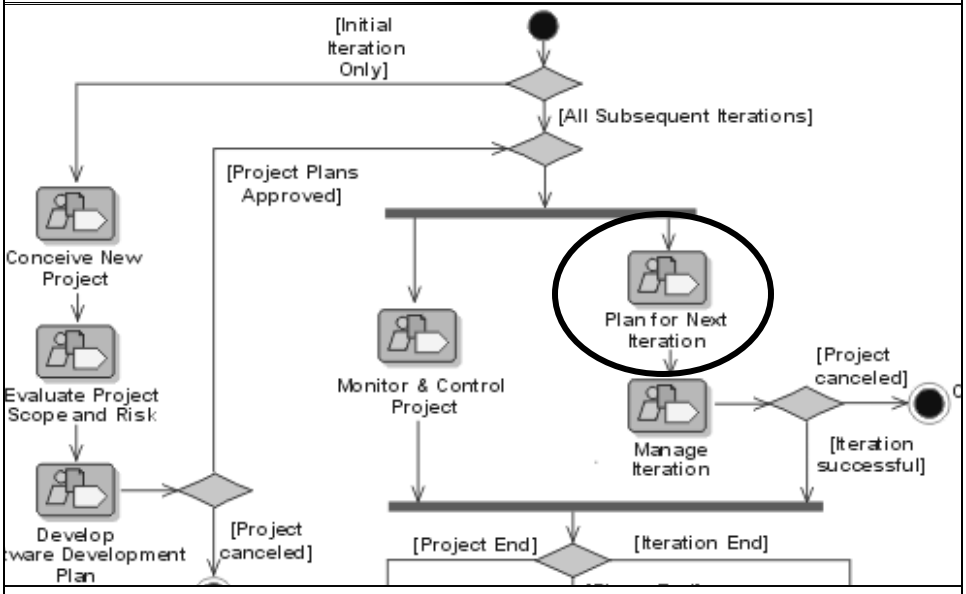


52

Copyright © Craig Larman. www.craiglarman.com

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

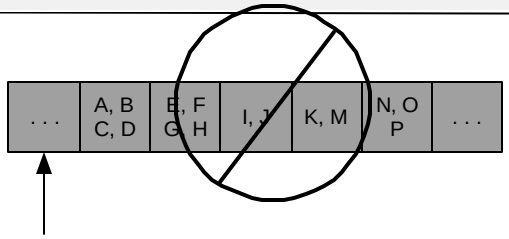
Adaptive, not Predictive Planning



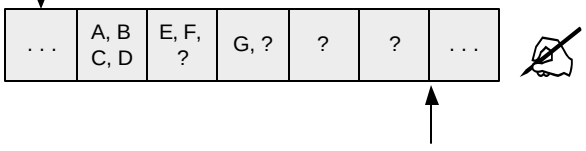
53

Copyright © Craig Larman. www.craiglarman.com

Adaptive, not Predictive Planning



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

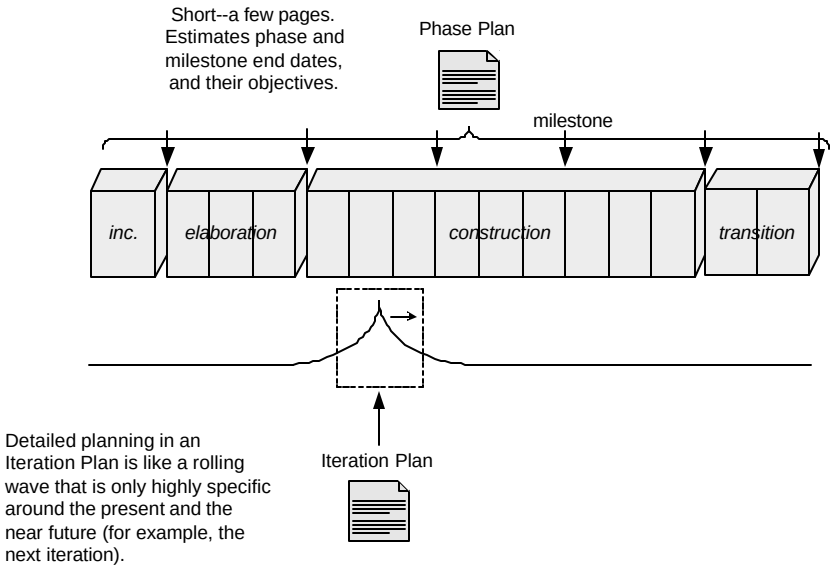


MILESTONE: A date (perhaps 2 months in future) and set of objectives is established.

54

Copyright © Craig Larman. www.craiglarman.com

Two Levels of Plans



55

Copyright © Craig Larman. www.craiglarman.com

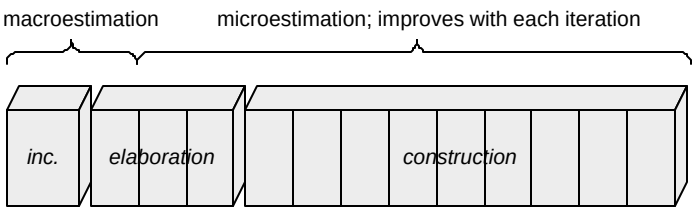
Iteration Plan

- Is *not* a plan of each iteration.
- What tasks, goals, requirements for the *next* iteration?
- At its simplest, just a list of goals or work.

56

Copyright © Craig Larman. www.craiglarman.com

Estimation and Phases



Estimates during inception are *not* used to commit to project duration and effort. Rather, they provide guidance to decide if it is worth continuing on to elaboration and doing some realistic investigation.

At the end of elaboration iteration 1, a believable estimate *starts* to emerge.

After 2 elaboration iterations, and more so by the end, there has been enough realistic investigation to generate and commit to overall project effort and duration estimates.

57

Copyright © Craig Larman. www.craiglarman.com

Environment

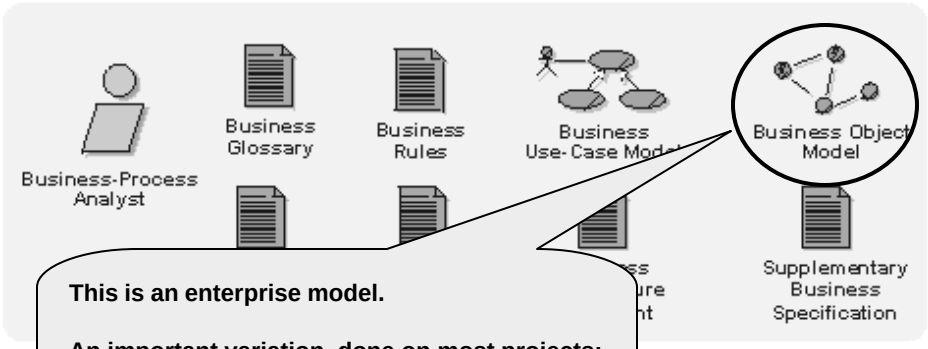


58

Copyright © Craig Larman. www.craiglarman.com

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

The UP Business Object Model



This is an enterprise model.

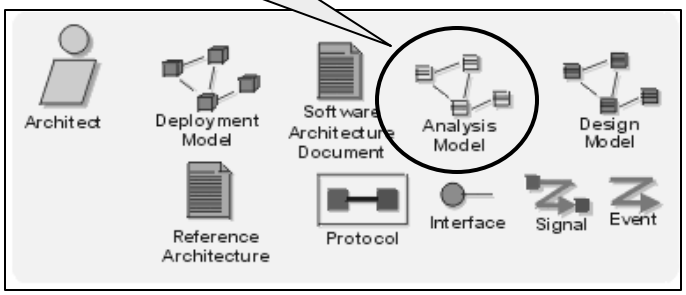
An important variation, done on most projects: the Domain (Object) Model. This is an application-level model. This is the one common artifact from Business Modeling.

59

Copyright © Craig Larman. www.craiglarman.com

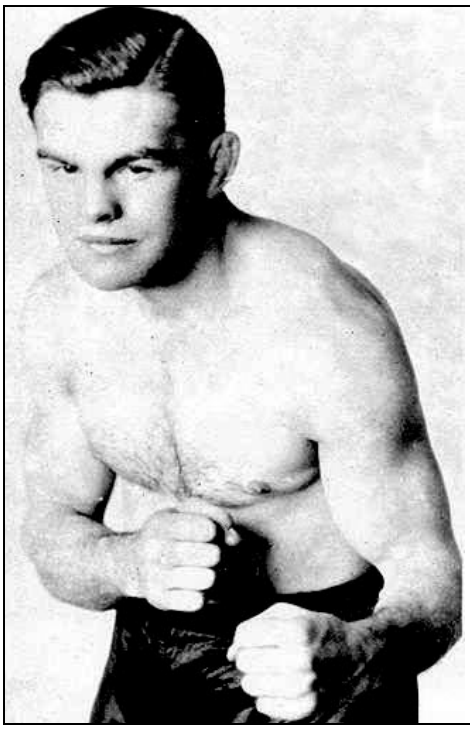
The UP "Analysis Model"

A high level generalization of the design model that omits fine details and provides a summary or abstraction of the design.



60

Copyright © Craig Larman. www.craiglarman.com



XP and Scrum within the UP

www.craiglarman.com
www.valtech.com

Copyright © 2002 Craig Larman. All rights reserved.

Scrum Practices within the UP

- Many are the same, or concrete realizations of UP practices.
  - Short time-boxed iterations
  - Iterative discovery and refinement of requirements and design
  - Adaptive, agile development

62

Copyright © Craig Larman. www.craiglarman.com

Scrum Practices within the UP

- Many are not explicitly mentioned in the UP, but are worthy of consideration, and complementary:
  - Daily scrum meeting
  - Management firewall
  - Common project room
  - Only by volunteering
  - Demo every iteration

63 Copyright © Craig Larman. www.craiglarman.com

XP Practices that we can use on a UP project?

- Many are the same, or concrete realizations of UP practices.
  - Short time-boxed iterations
  - Iterative discovery and refinement of requirements and design
  - Adaptive, agile development
  - Continuous verification of quality —> test first development

64 Copyright © Craig Larman. www.craiglarman.com

XP Practices that we can use on a UP project?

- Many are not explicitly mentioned in the UP, but are worthy of consideration, and complementary:
  - Common project room
  - Continuous integration
  - Only by volunteering
  - On-site customer
  - Pair programming, …
- But a few are incompatible or problematic …

65 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Written Requirements, or Docs?

- Aiming for simplicity and fast programming, XP discourages the creation of written documents, such as
  - the UP use cases
  - the UP Supplementary Specification (e.g., nonfunctional requirements)
  - the UP Software Architecture Document

66 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Written Requirements, or Docs?

- While sympathetic to the ideal of avoiding "write-only" documents, many do not find it practical to run or hand over a consulting project without such artifacts.
  - And it is (unfortunately) rarely possible to get an on-site customer.
  - The XP "cards" are not sufficient on larger, distributed projects, or those where continual involvement by subject matter experts is not possible.

67 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Design and Diagramming Work?

- XP suggests "a very short period" of design thought (including diagramming, if any), before programming starts in an iteration.
- Yet, one of the best practices of the RUP is to "model visually."

68 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Design and Diagramming Work?

- Also, a large portion of the brain is “optimized” for visual (iconic and spatial) languages.
  - Visual languages such as the UML play to a natural strength of the brain.
- While XP does not deny the use visual languages, it minimizes their use to very short periods (e.g., 10 minutes) before programming.

69 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Design and Diagramming Work?

- In contrast to the XP approach, in the UP, on a 2-week iteration (for example), something like a ½ day or full day of design speculation and diagramming near the start of the project is considered appropriate.
  - It is not a massive design and diagramming effort, but it is not “10 or 15 minutes”.

70 Copyright © Craig Larman. www.craiglarman.com

XP Inconsistencies with UP: Priority of Work, and the Phases

- XP emphasizes customer-driven work priorities.
- The UP emphasizes doing the riskiest things first, in order to drive down the risky element early and increase the chance for project success.
  - "Fail-early" scheduling.
  - Risk-driven scheduling gives rise to the elaboration phase.

71 Copyright © Craig Larman. www.craiglarman.com

It's More Like a UP Project than an XP Project, if…

- Short timeboxed iterations.
- Iterative discovery and refinement of the requirements and design.
- Use cases are written.
- ½ or 1 day is spent on some design thought and diagramming near the start of a 2-week iteration.
- A Supplementary Specification and Architecture Document are written.
- Test-first programming.
- Risk-driven iterations, with an elaboration phase.
- Continuous integration.
- Common project room.

72 Copyright © Craig Larman. www.craiglarman.com

UP Weaknesses

- Lacks the attention to social and psychological factors of XP.
- So many optional activities and artifacts and guidelines, one can miss the forest for the trees.
  - And, "fill in the boxes" syndrome.
- Urge to superimpose waterfall phases on UP phases.

73 Copyright © Craig Larman. www.craiglarman.com

Open Questions & the XP

- Kent Beck is speculating that the curve of late-change has gone significantly flatter in recent years (and thus, avoid design for flexibility/protection with respect to speculated product evolution).
- Barry Boehm and Caper Jones have done research and can back their assertions with data to the contrary from multiple projects. XP can't. Maybe the XP assertion is true. Nobody really knows.
- Clearly there is such a thing as too much speculative design and too much over-engineering.
- But there is also too little, so that the cost of change is higher than if forethought was applied.
- Bottom line: Research/proof is required; not claims.

74 Copyright © Craig Larman. www.craiglarman.com

Open Questions & the XP

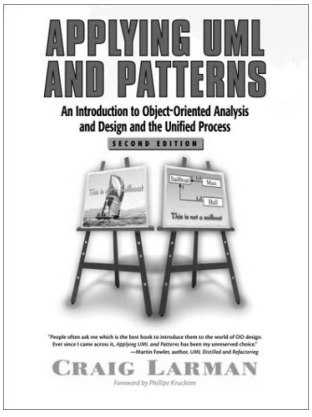
- The value of visual modeling and design before programming?
  - "50%" of brain dedicated to visual processing.
  - Visual languages abstract details, and allow rapid change and communication.
- Different people have different cognitive styles and strengths.
- Clearly there is such a thing as too much diagramming or speculative design. But I claim there can also be too little.
  - In a 2-week iteration, a claim of, for example, "10 or 15 minutes on Monday is better that 5 hours" is speculative.
  - Nobody knows. Research is required.

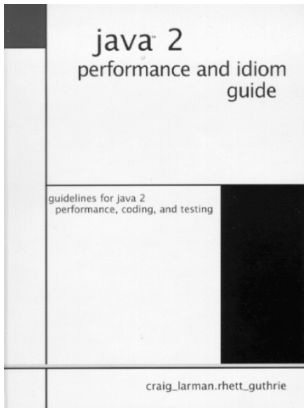
75

Copyright © Craig Larman. www.craiglarman.com

speaker background

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





Underway:

Agile and Iterative Development: A Manager's Guide

76

Copyright © Craig Larman. www.craiglarman.com