290T: The Business of Software: Teams and Processes in SW

Professor Kurt Keutzer
Fall 2003
EECS
keutzer@eecs.berkeley.edu

Class News

- **For case on Wednesday:**
  - **John Fernandes, Director of Business Development and Strategy in the Emerging Business Team, Microsoft**
- **Before Monday: Please take the personality inventory at:**
  - **http://www.humanmetrics.com/cgi-win/JTypes1.htm**
  - **And record your results**

Due today: Interim results on your development of the Business Model, Template was presented in Lecture 7

- **Team overview**
- **Business environment**
- **Product Overview**
- **Marketing Plan**
- **Distribution strategy**

Outline

- **WRS retrospective**
- **Review of elements of software process**
- **Putting elements of software process together**
- **Key points of software project management**
- **Reflections on teams and software**

3

What did we learn from Jerry?

4

Issues WRS Faced as they Grew

- Evolving management and organizational structure
- Mergers and acquisitions
- Changing trends
  - End-customer requirements
  - Technology
  - Competitive environment
  - Wall street

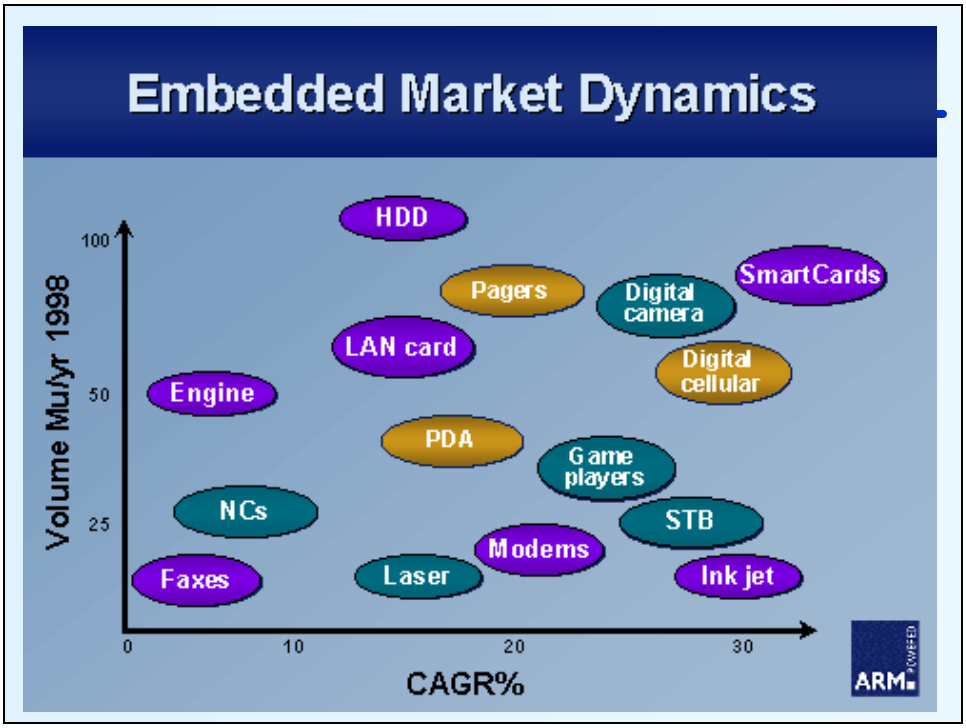
5

Proliferation of Embedded Devices

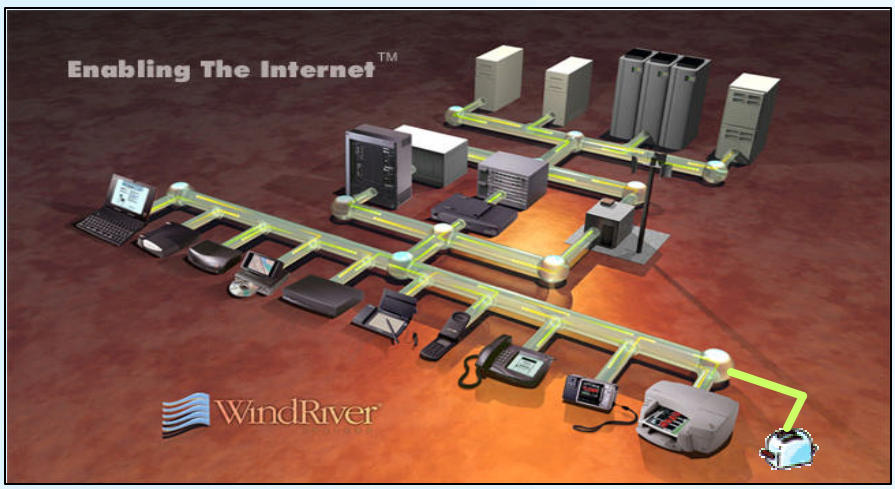
- Most of our access to information will come through embedded systems

Jerry Fiddler, DAC Keynote 2002

6



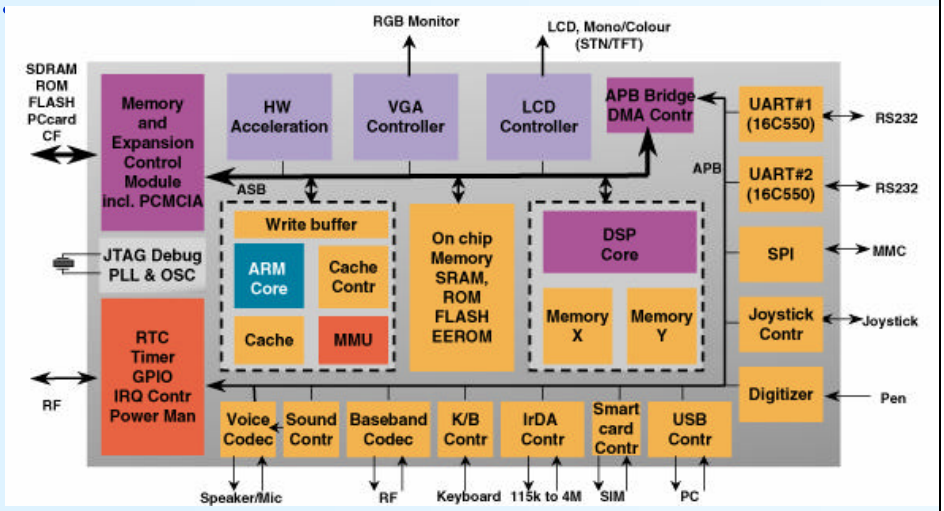
Everything is connected ....



Jerry Fiddler, DAC Keynote 2002

8

embedded system-on-achip



Proliferation of peripherals/device drivers/code complexity

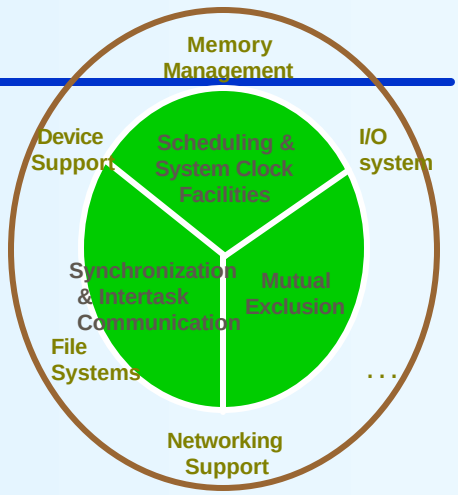
Courtesy Synopsys

9

WRS Golden Era

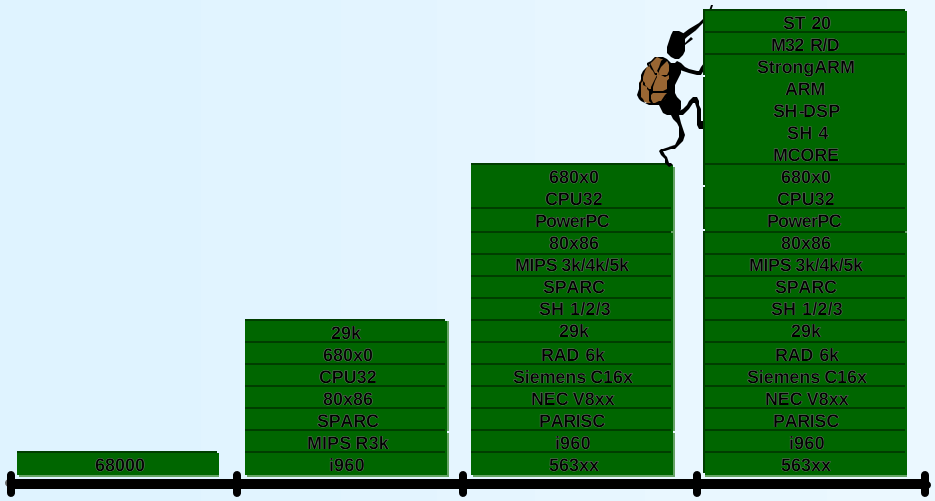
- **Wind River System enjoyed it's "gold era" when:**
  - **A single application language – ANSII C**
  - **Demanding, but manageable IDE and RTOS requirements**
  - **>1, but still a modest number of target processors**

29k
680x0
CPU32
80x86
SPARC
MIPS R3k
i960



10

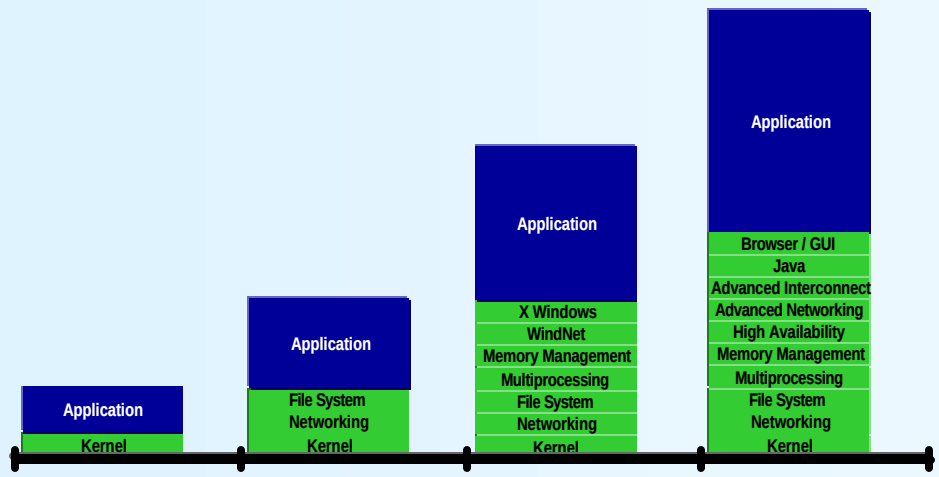
Microprocessor Chaos



Jerry Fiddler, DAC Keynote 2002

11

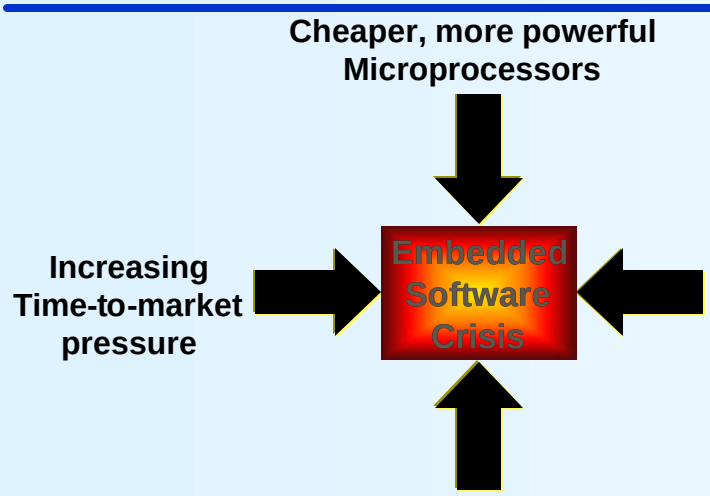
End Customer: Software complexity increasing....



Jerry Fiddler, DAC Keynote 2002

12

Embedded Software Crisis

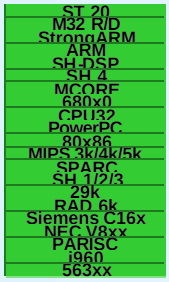


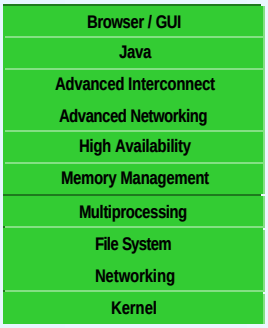
Jerry Fiddler, DAC Keynote 2002

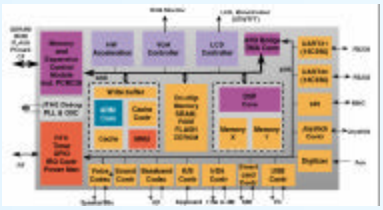
13

WRS Challenging Era

- It's a challenging time for WRS
  - More variety of application languages: C, C++, Java, Matlab …
  - Increasing number of IDE and RTOS requirements
  - Tremendous proliferation of target processors to support







14

Plus Increasing Competition

- **VxWorks : Wind River Systems**
- **OS-9 : Microware**
- **OSE : Enea Systems**
- **EPOC : Symbian**
- **VRTX : Mentor Graphics**
- **Windows CE : Microsoft**
- **Palm OS : PalmSource**
- **Spox :(Spectron Microsystems)**
- **HP-RT : Hewlett-Packard**
- **Integrity : Green Hills**
- **Linux, RTLinux : FSMLabs etc.**

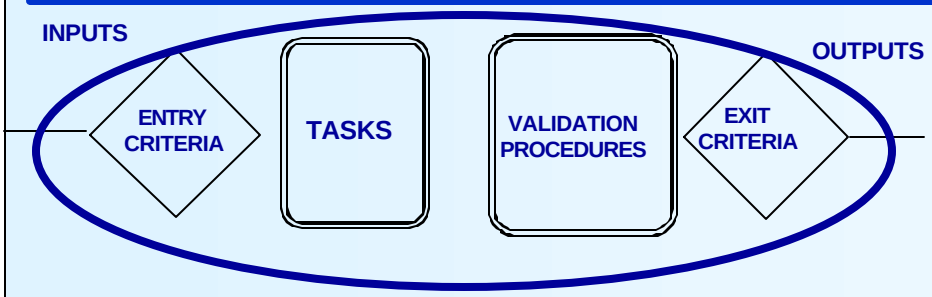
15

Outline

- **WRS retrospective**
- **Review of elements of software process**
- **Putting elements of software process together**
- **Key points of software project management**
- **Reflections on teams and software**

16

One Step of a Process



- **Clearly defined:**
  - **Entry criteria – can we start?**
  - **Repeatable tasks – what do we do?**
  - **Validation – how do we know we did it?**
- **Exit criteria – are we done?**

17

Process Documentation Template

PROCESS DOCUMENT SECTION	OBJECTIVES
1. Objectives	What are the objectives for this element of the process? what does it try to achieve?
2. Owner	Who is the owner for this element of the process? who is responsible person for the documentation, communication, maintenance, and continuously improving this process?
3. Inputs	What are the inputs? where are the inputs coming from? are there any constraints or dependencies for the inputs?
4. Outputs	What are the outputs? where are the outputs going to? are there any constraints or dependencies for the outputs?
5. Entry Criteria	What are the starting requirements for this element
6. Exit Criteria	What are the completion requirements for this element?
7. Tasks	What are the tasks required to achieve the objectives of this element?
8. Dependencies/constraints	Are there any requirements/constraints for the tasks or steps in the process? Who are the owners for the dependencies?
9. Validation	What are the measurements for the process? How to know that tasks work as expected? How to know that objectives are met?

18

Dreaming

- Inputs: problems
- Tasks:
  - Reflection, typically by an individual or an informal team (e.g. after softball practice):
    - Could the idea work (technology)?
    - Could it make money (market)?
- Validation:
  - Bouncing the idea off of friends, mentors etc.
- Output:
  - a new technology (logic synthesis)
  - a new market opportunity (Matlab acceleration)

19

Investigating

- Inputs: a new technology/a new market opportunity and questions
- Often the key question in a marketing driven idea is a technical question. Alternatively , the key question in a technically driven idea is a market question.
- Tasks:
  - Technology: Could we really accelerate Matlab?
    - Key issues:
      - Can we perform “compile code” simulation for Matlab? Or do we have to do interpreted-code Matlab simulation?
      - Key point: managing types in Matlab)
  - Marketing: Is the market for my bright idea big enough?
    - Prove in key market issues:
      - Market segmentation
      - Market sizing: top down, bottom up
      - Customer interviews
- Output: Confirmation of technology or market – a “business case”

20

Requirements specification

- **Inputs: target customer and market**
- **Tasks:**
  - **Market requirements:**
    - **What are broad characteristics of the market? (How much will Matlab customers pay for an accelerator?)**
  - **Customer requirements:**
    - **What are specific customer requirements? (How much acceleration is required to be interesting? 2x 10x?)**
    - **Must "look and feel" remain the same?**
    - **Speed versus accuracy trade-offs?**
- **Output: Market requirements specification**

21

Software Design specification

- **Inputs: market requirements specification**
- **Tasks:**
  - **Group market requirements:**
  - **For each identified market market requirement**
  - **Translate into major programming task**
  - **Group programming tasks into a software architecture**
- **Output: Design specification including hierarchical software architecture**

22

Implementation and Test Plan

- **Inputs: Software Design Specification**
- **Tasks:**
  - **Refine major software modules in architecture into manageable (for the programmer) and schedulable (for the manager) software tasks**
  - **Create unit test plan for each individual unit**
  - **Create system test plan for system and validation in environment**
- **Output: Implementation and test plan and targets for acceptance**

23

Other software activities

- **Reviews of all of the above**
- **Hiring/recruiting**
- **Actual implementation**
- **Actual unit testing**
- **Actual system testing**
- **Porting**
  - **SW needs to be tested on all the platforms (PC variants, Workstations: SUN, SGI, HP) on which it runs**

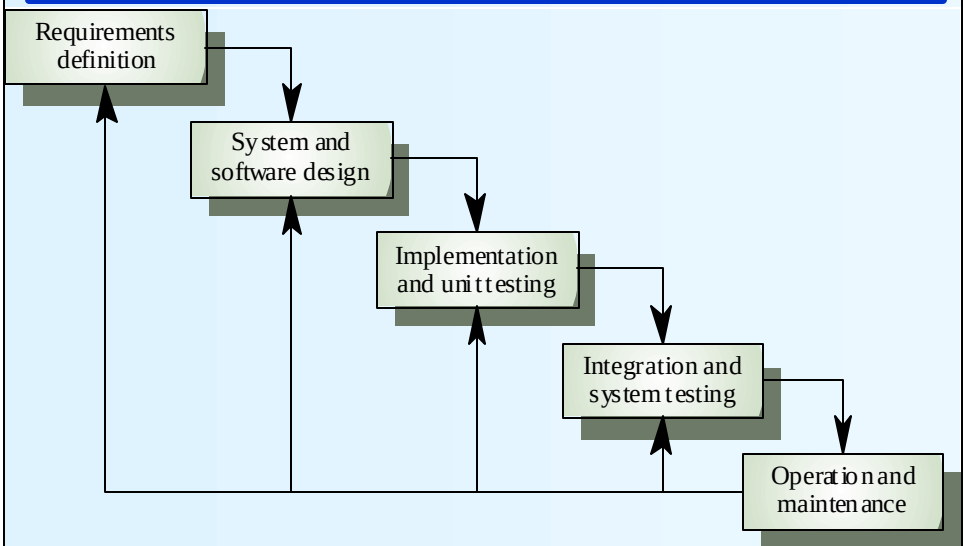
24

Outline

- **WRS retrospective**
- **Review of elements of software process**
- **Putting elements of software process together**
- **Key points of software project management**
- **Reflections on teams and software**

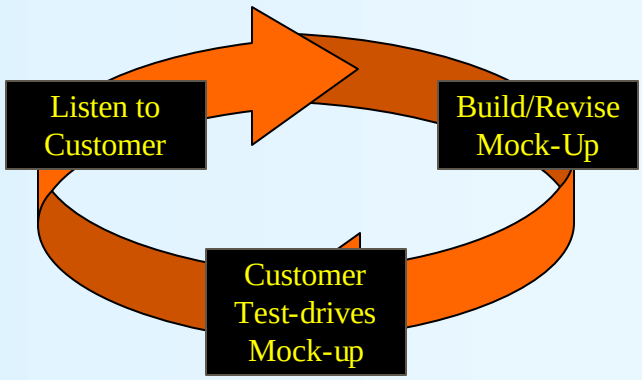
25

Waterfall model



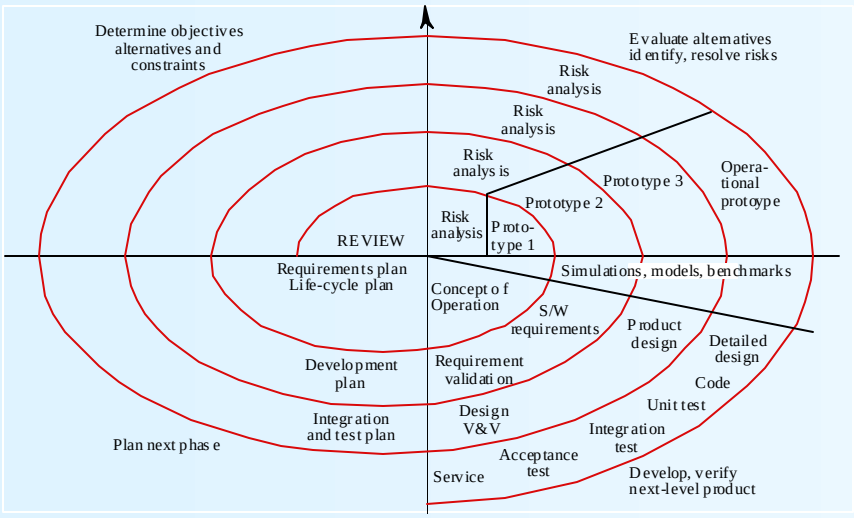
26

Prototyping Model



[Pressman 97]

Spiral model of the software process



Outline

- WRS retrospective
- Review of elements of software process
- Putting elements of software process together
- Key points of software project management
- Reflections on teams and software

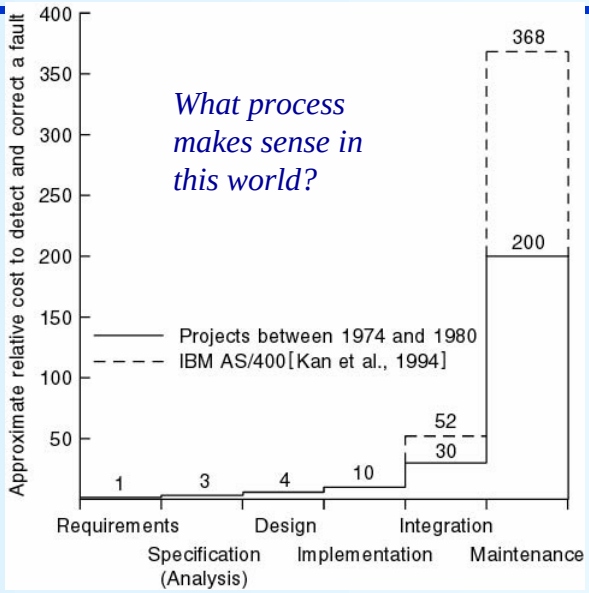
29

So which process?

- Common elements:
  - Gather requirements before you implement
  - Develop acceptance criteria and test as you go
    - Know when to pull the plug on a project
  - Provide mechanisms for iteration - even back to the beginning – you may need them anyway
  - Provide mechanisms for direct customer feedback before actual release
    - Beta-testing in structured development
    - Phased-releases
- Fit process to your software problem

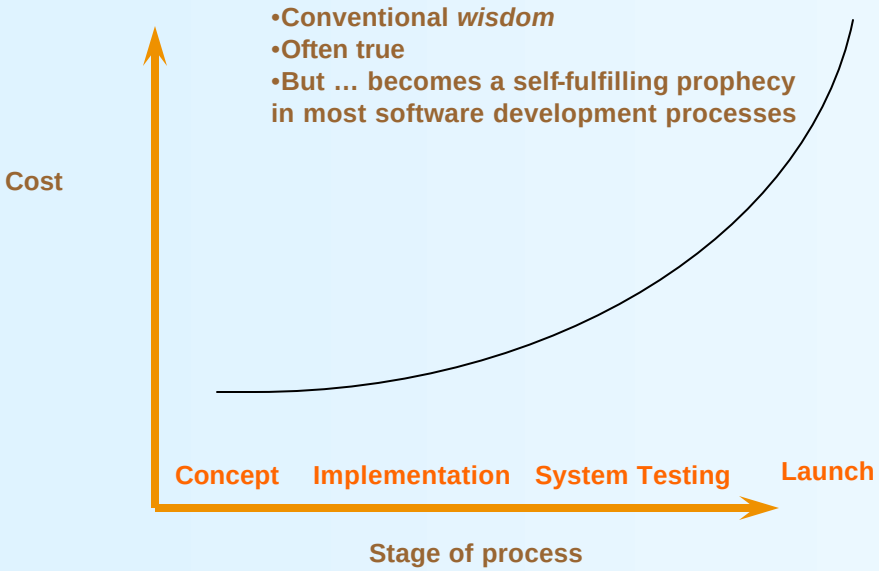
30

Cost to Detect and Correct a Fault



31

Premise: Cost of change

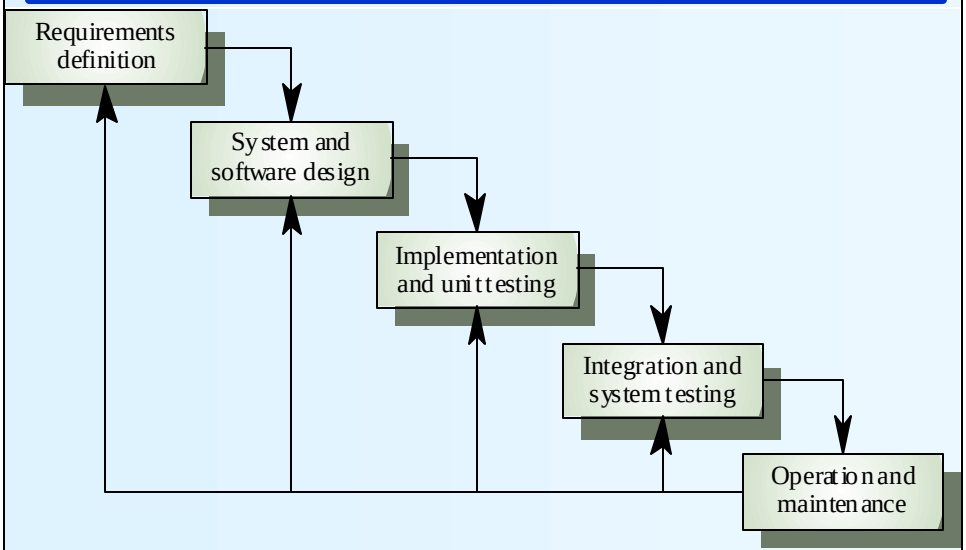


32

Given these facts which process makes sense?

33

Waterfall model



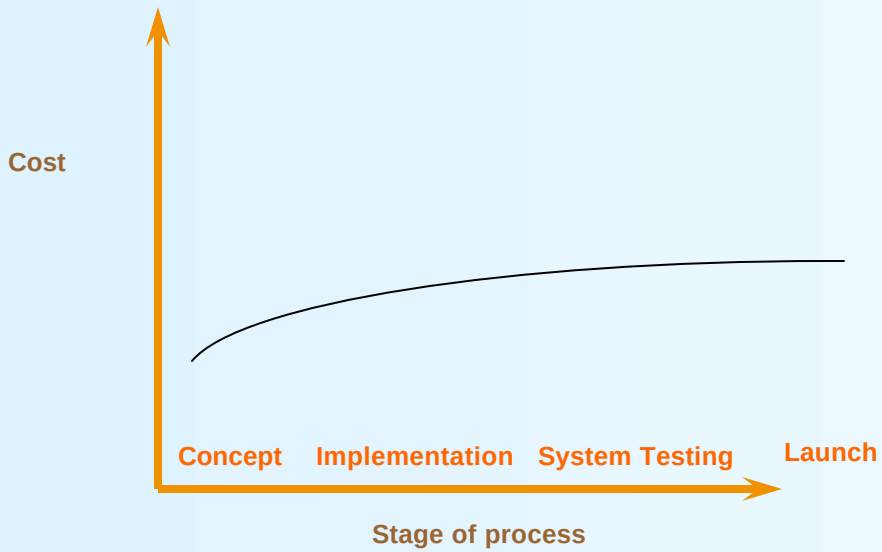
34

Cost of a Design Change

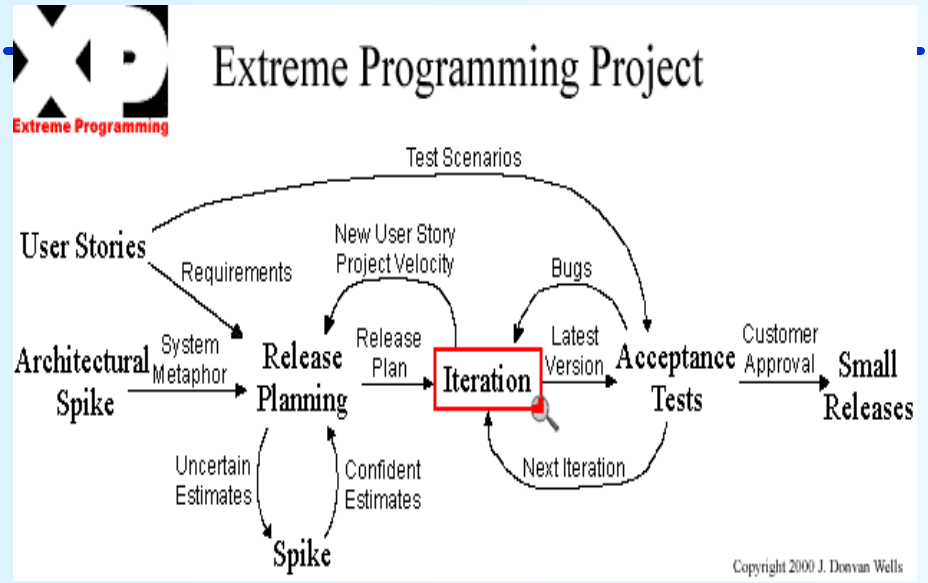
- Many organizations have created slow laborious software design processes
- In such cases *of course there is a high cost of design change*
- Bad reasons not to change the design – even in the field
  - Diffusing change expensive – silly in ``communication age''
  - GUI change could confuse customer – current GUI may be even more confusing
- Good reasons not to change the design
  - Life critical situations
  - Situations requiring significant retraining
  - Customer perceptions, etc. ..

35

What if?



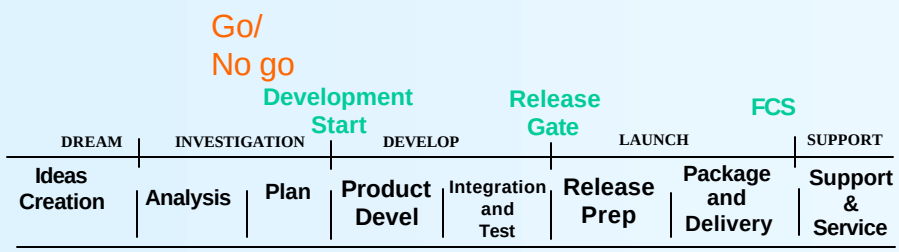
36



Choosing a Software Process

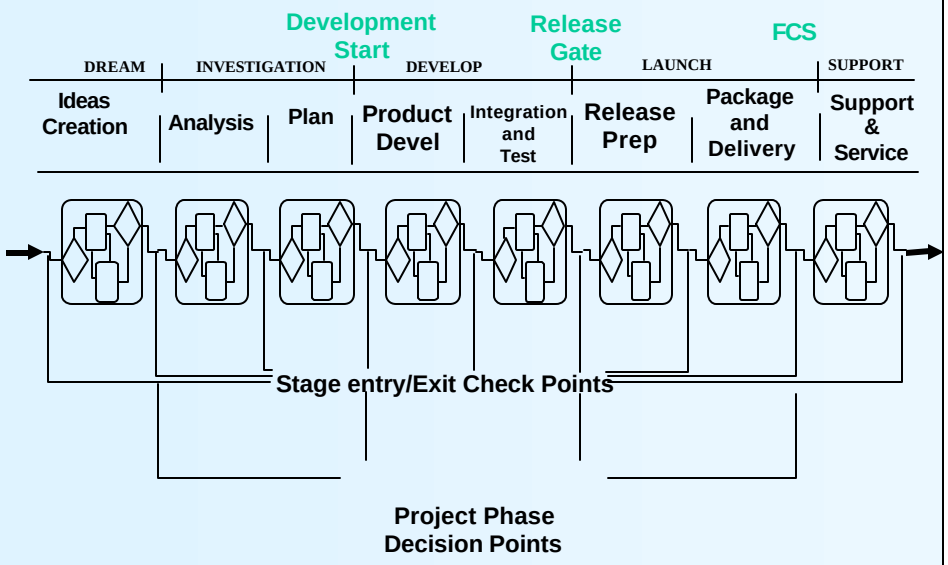
- **Fit the process to the software problem**
  - **What are the most important characteristics of the problem?**
    - **Time to market (TTM)?**
    - **Robustness?**
    - **Cost of development?**
- **Administrate the process faithfully**
  - **If you do waterfall process – generate documents**
  - **If you do extreme programming – generate iteration releases**
  - **If you do ``Synopsys'' hold the review meetings**

Where do the biggest slips occur?



39

SNPS Software Development Process



40

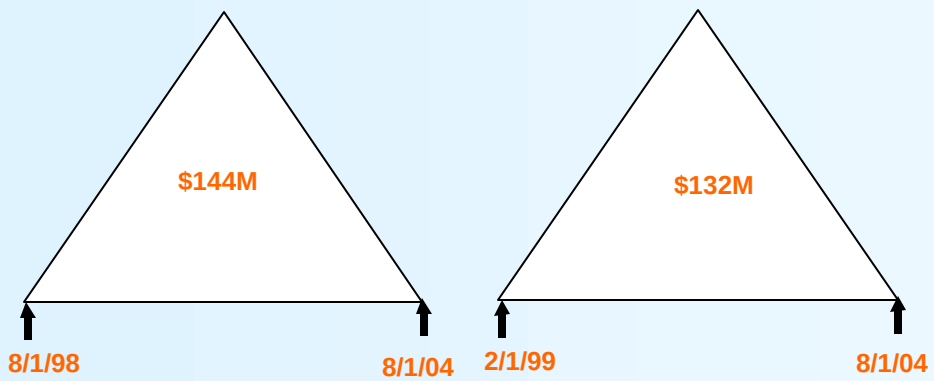
Outline

- **WRS retrospective**
- **Review of elements of software process**
- **Putting elements of software process together**
- **Key points of software project management**
- **Reflections on teams and software**

41

Managing the product development cycle

- **Impact of 6 month development slip in 6 year product cycle?**

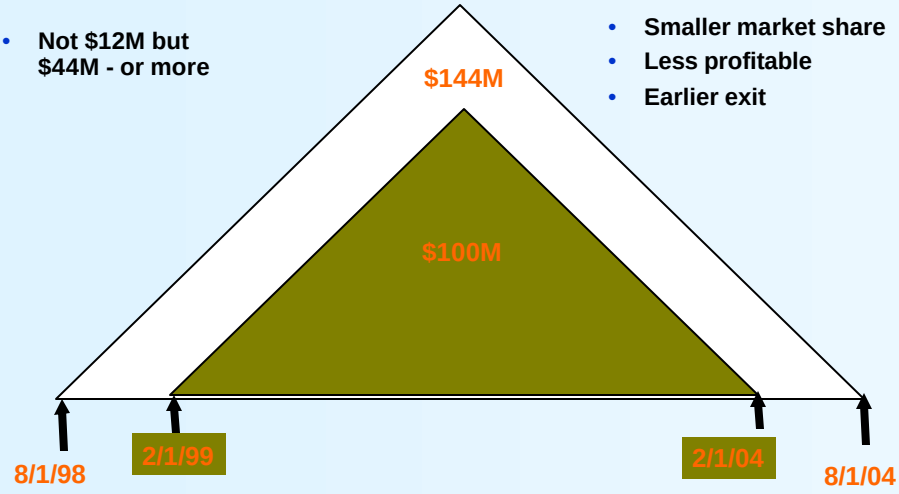


42

More likely impact of 6 month development slip

- Not $12M but $44M - or more

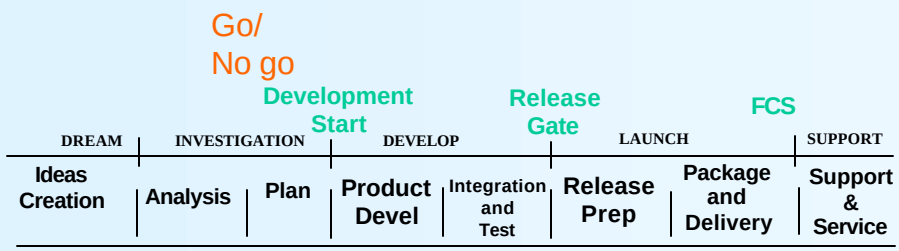
- Smaller market share
- Less profitable
- Earlier exit



$250K please

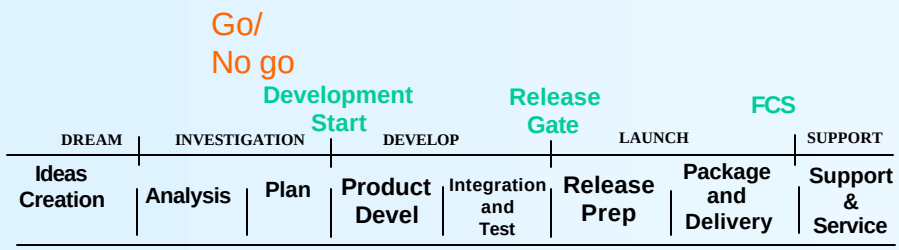
43

Where do the biggest slips occur?

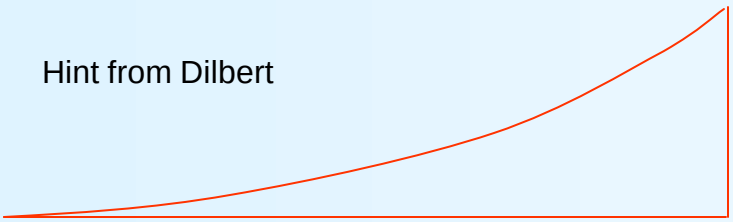


44

Where do the biggest slips occur?

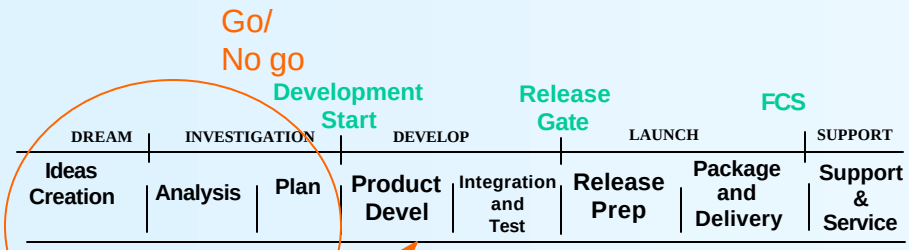


Hint from Dilbert



45

Must Front-load management attention



- Dilbert is right
- Management focuses on the wrong part of the process

46

In an informal analysis

- **In an informal (e.g. hard to get hard data) analysis of 25 products at Synopsys we (Bob Dahlberg and I) found:**
  - **Most time (often >2 years) was spent in ``investigation phase''**
  - **Any product that was ``late'' to market by more than 6 months**
    - **VSS – VHDL simulator**
    - **FPGA express**
    - **Chip architect – chip floorplanner**
    - **Behavioral synthesis**
  - **Had been *under investigation for at least 2 years***
  - **All these products came under *intense management scrutiny* when they were finally in development**
  - **Rapid management decision making would have given more than adequate time for development**

47

Rapid decision making

..most companies don't die because they are wrong; most die because they don't commit themselves They fritter away their momentum and their valuable resources while attempting to make a decision. The greatest danger is standing still.

Andy Grove "Only the Paranoid Survive"

48

Outline

- **WRS retrospective**
- **Review of elements of software process**
- **Putting elements of software process together**
- **Key points of software project management**
- **Reflections on teams and software**

49

Summary

- **Initiating and managing a successful software product entails:**
  - **Recruiting and staffing the team**
  - **Clearly designating the proper roles and responsibilities**
  - **Identifying the objective of the software product**
  - **Choosing a software development process that meets those objectives**
  - **Managing the process**

50

Individuals and Group Dynamics

- Jung – psychological foundation
- Myers-Briggs – modern theory
- Keirsey etc. – broad dissemination www.keirsey.com
- Last ``free'' site: http://www.humanmetrics.com/
- The key dimensions
  - Introversion (I) vs. Extroversion (E)
  - Intuitive (N) vs. Sensing (S)
  - Thinking (T) vs. Feeling (P)
  - Perceptive (P) vs. Judging (J)

- There's no right and wrong and you can't really change yourself anyway!
- Learn the strengths and weakness of your character type

51