Systems Development and Implementation

Lesson 5

http://faculty.washington.edu/blommers/syllabus_sys_dev_imp.htm

This Evening in Seattle

- Part I - Applications of Interviewing
  - ✓ Lecture ... done
  - ✓ Questionnaires ... done
    - Team interviews to be done
    - Present core requirements to class
- Part II -Test Plans (background activity)
- Part III - Product Evolution and Testing

Part I
Interview for Requirements

Team Exercise

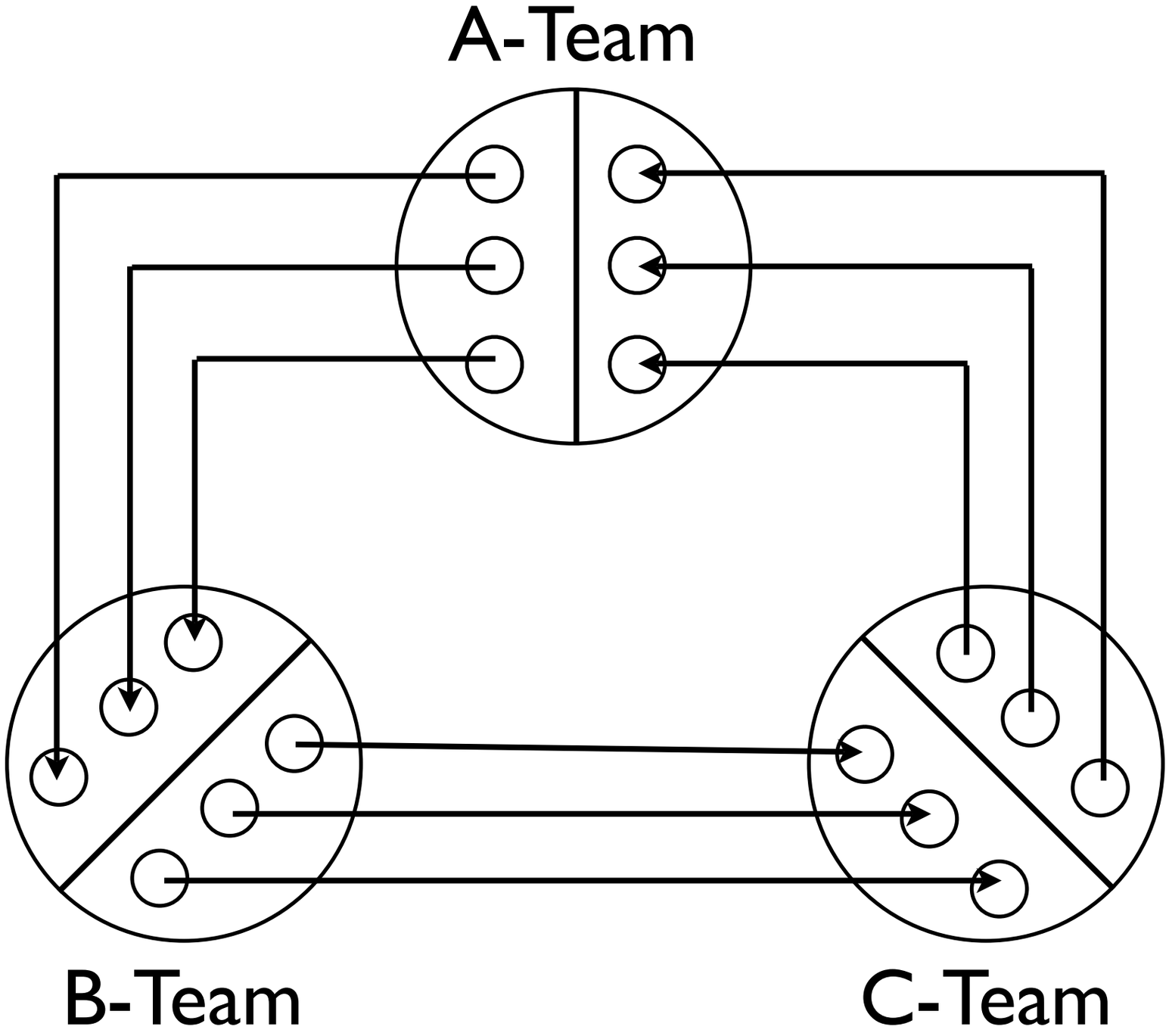
- Review the needs statement written on the board
- Create a questionnaire to find requirements
- Conduct 1-1 interviews to gather requirements
- Combine interview results
- Present requirements back to “management”
- Make any changes to the requirements
- Present and compare these to “management”

A-, B-, C-Team

What is your team's needs statement?

Write it on the board for reference.

Interview Relationships



Part I - Applications of Interviewing

Interviewing is an interpersonal skill. Its objective is to draw unbiased high-quality information from a subject matter expert in a one-on-one setting. The interview lasts 30-90 minutes. The information collected might be applicable to any one of the SDLC stages. While interviews are often used to gather project feature requirements, they may just as well be applied to defining prototype features or generating test cases.

Interviewing

- Begin with the needs statement:
  - "Acme Insurance needs a secure remote access system to allow our clients with large automobile fleets to directly access all their outstanding claims online in real time from any location (because ...)."
- Acme is an SMB with a tiny IT staff and will hire a consultant to develop requirements and report back on a range of potential solutions
- Today we will focus on testing requirements

Consultant's Tasks

- Arrange to interview Acme stake-holders and SMEs
- Develop a questionnaire prior to the interviews
- How do we develop this questionnaire?
    - You have to know about networks & security!
    - Pose open ended questions - "Tell me about ..."
    - Who, what, where, when, why, how
    - "Is there something else I should be aware of?"
- Schedule and conduct one-on-one interviews
- Organize results of the interviews

Two Types of Questions

- Closed-ended questions limit the amount of information they gather to a limited number of pre-defined nuggets - yes/no/100
- Open-ended questions allow the interviewee to offer information that the interviewer may not expect, and that can be most valuable.
- "Do you feel bad about that?"

  versus
- "How does that make you feel?"

Consultant's Tasks

- Present the discovered (testing) requirements back to responsible management
- Each requirement may be modified or rejected
- "X is not a requirement. Fred always wants that."
- Undiscovered requirements may be added
- The final requirements are agreed upon bindingly
- Final requirements drive the analysis phase

Acme Interview Questions

- How many clients do you have now? In the future?
- How often do clients login, for how long, are there any patterns (Friday, month end ...)?
- From where will your clients login? (language issue?)
- Describe all the transactions they'll perform.
- What kind of support will your clients expect?
- Will clients call your help desk? (Closed ended Q)
- What range of security solutions are appropriate?
- How will the total solution be managed?

Consultant's Tasks

- Consultant does SDLC analysis and design
- Develops multiple alternative solutions
- Solutions differ in cost, complexity, scaleability, longevity, support, technology, and goodness of fit to the requirements
- Present solutions to management
- Note how important it is to have a fixed set of agreed-upon requirements

Definitions

Testing requirements flow back up from features and derived requirements. Anything being built has to have a test plan.

Gathered estimates for transaction parameters

"Print Open Claims" transaction

30 clients

56Kbps Worst case connection speed

100 per day at month end

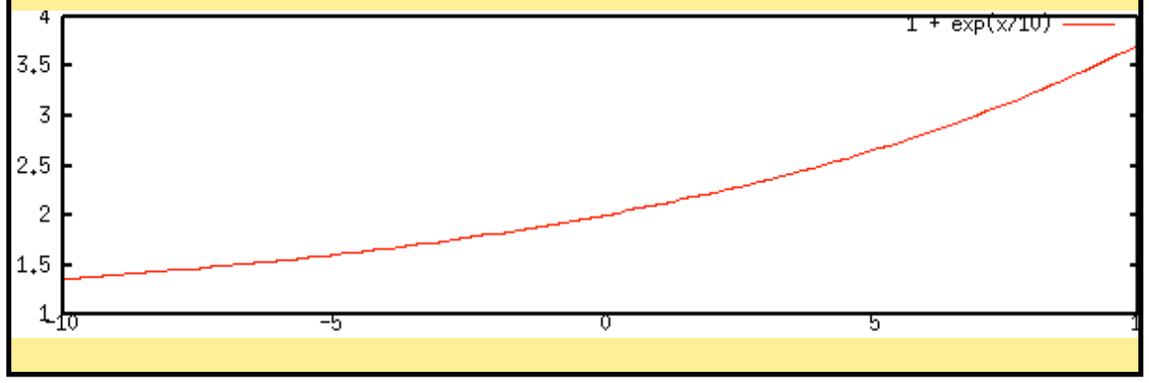
Testing plans flow from the testing requirements

Measure transactions in a benchmark or pilot

Max 10 CPU seconds per transaction

100KBytes each

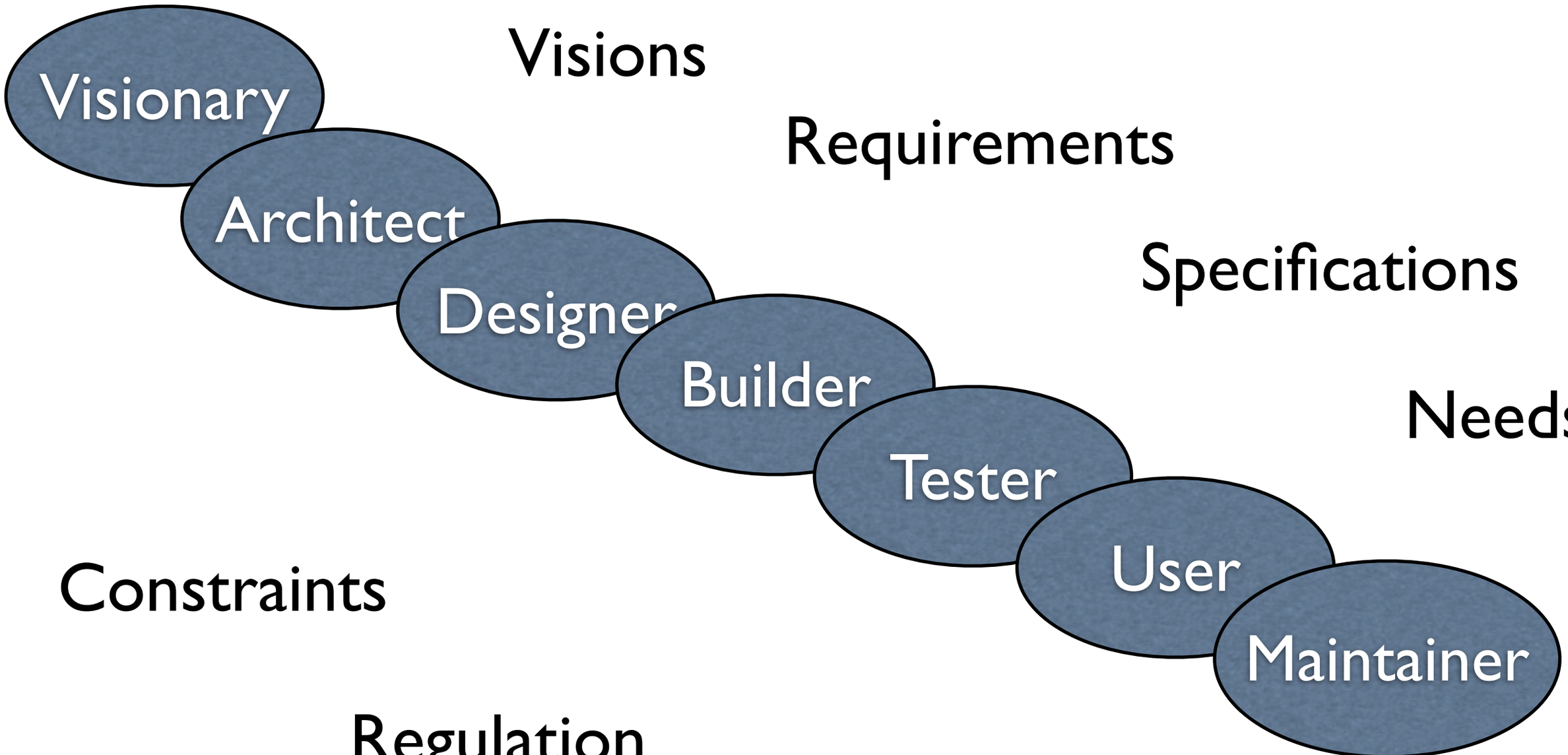
Model and simulate to verify scalability



Part II - Test Plans

- Cause of errors in an implementation:
  - features are not well-understood
  - requirements are incomplete or wrong
  - logical design (architecture) is flawed
  - design errors
  - implementation is flawed (bugs)
  - test cases are not comprehensive
- Blame it on RAD?
- Blame it on ignorance and inexperience?

People View of SDLC



The SDLC, our old Friend

Phase		Activity	What you have when you're done
1	Identify	Identify unmet business requirements. (ex JAD)	Business feature requirements, justification, broad scope. "What"
2	Initiate	Scope out the system and project plan	Costs, staffing, timetable (Gantt), critical success factors, locations effected, participants. "Who, when, how much"
3	Analyze	Requirements analysis	Comparison of alternative solutions. Testing requirements. "How, why, where"
4	Logical Design	Architecture - blocks of functions - data movement	Functional details about data, inputs, outputs, models, diagrams, charts. "How, what"
5	Physical Design	Technical details	Technical specifications about networks, storage, compute power. "What, how"
6	Prototype	Testing trial with partial functionality. (ex RAD)	Lessons learned about functionality, scalability, usability, reliability, fine-tune implementation schedule. Test results. GoTo steps 1-5 as needed to apply the lessons.
7	Implement	Real hardware, software	System is fully operational. Awards & accolades
8	Maintain	Support, repair, enhance	This never ends until the system is obsolete, cannot be upgraded, and is replaced

About Testing

- Functional (check each feature out)
- Performance (automated script-driven testing)
- Reliability (see long list on next slide)
- Note that good requirements include testing hooks, debug code, logs, tracing ...
- Hire n00b testers (developers suck at testing)
- Ease & efficiency of use (any interface)
- Can the user discover available features (s/w)
- Inspection (construction trade)

Validating Requirements

- Is the requirements statement concise?
- Complete? Accurate? Consistent? Testable?
  - MUST and SHALL words
  - SHOULD words
  - COULD or MAY words
  - Informative: clarifying informational examples that can be removed without loss
  - Normative: contract language that must be adhered to

Requirements

- performance
- security
- scalability
- manageability
- maintainability
- reliability
- cost
- risk
- time table
- supportability
- expandability
- power
- air
- light
- heat
- operations
- funding
- staffing
- quality
- legal
- education
- training
- organization
- environmental
- documentation
- interface
- storage
- transmission
- connectivity
- network
- hardware
- software
  - apps
  - servers
  - drivers
  - licenses
- support
- functions
- features
- and
- so
- on

Decomposition

who, what, where, when, why, how

Brainstorm to generate more areas

Do we have testing requirements in these areas?

Test all aspects of operations

Testing Alternatives

- Follows might be called trade studies:
  - Which typewriter keyboard layout is best?
  - Which type of lamp filament burns longest?
  - Should the mouse have 1, 2, or 3 buttons?
  - Is a doorknob better than a vertical handle?
  - Is a second video output needed?
- We don't have the answers and we can get them during the pilot phase

A-, B-, C-Team

Describe the pilot project.

Present your pilot project test plan.

Part III -Product Evolution and Testing

- Personal Writing System
- Removable Disk Storage
- Television Set
- Other examples

Testing Issues

- Generational product evolution means the products are constructed so differently and have such radically new feature sets that new types of testing are needed. Testing changes as products do.
- In the following examples, products get more capable with each generation. At some point old tests stop making sense and brand new ones are needed.

A-Team B-Team C-Team

- Objective: Each team will learn about testing evolution by working one of examples

Personal Writing System

Gray goose quill dipped in ink

Steel pen and ink (Shaeffer)

Ball-point pen (Bick)

Typewriter (Underwood)

Dedicated word processor (Wang)

vi (a Unix text editor)

LaTeX (Donald Knuth's typesetting language)

Early word processor (Scripsit, Wordstar)

Modern word processor (KWord, Abiword)

- Handwriting
- Keyboarding
- Electronic typesetting
- WYSIWYG

- Liquid Ink
- Packaged Ink
- Inked Ribbon
- Cartridges
- Impact printer
- Laser printer
- Inkjet printer

A-Team

Testing Issues

- Develop a test suite for each personal writing system
- Organize the tests into columns
- What pattern do you detect as the product evolves?

Product	Test 1	Test 2	Test 3	Test 4		
	Duration of ink charge	Cost	MTTF	Productivity		
Quill	1 lines	$ 20	20 pages	10 ppd		
Steel pen	10 lines	$ 10	100 pages	20 ppd		
Ball-point	100 pages	$ 1	100 pages	20 ppd		
Typewriter	200 pages	$ 200	10,000 pages	100 ppd		

Removable Disk Storage

- 8-inch Floppy disk (~80KB)
- 5-1/4 inch Floppy disk (~100KB))
- 3-1/3 inch Floppy disk by Sony (1.44MB)
- ZIP Disk from Iomega (100MB)
- Superdisk (120MB)
- CD (700MB)
- DVD (4.7-8.5GB)
- HD DVD (15, 30GB)
- Blu-Ray (25, 50 GB)

- Magnetic surface
- Optical pits
- Dual layers
- Shorter wavelengths
- Rotation speed
- Application
- Controllers
- Disk size
- Daisy chaining

B-Team

Testing Issues

- Develop a test suite for each removable disk system
- Organize the tests into columns
- What pattern do you detect as the product evolves?

Product	Test 1	Test 2	Test 3	Test 4		
	Capacity	Cost	Error rate	Useful Life		
8-inch	80 KB	$ 20	10^-5	100 ops		
5-1/4 inch	100 KB	$ 5	10^-6	200 ops		
3-1/2 inch	1.44 MB	$ 1	10^-7	400 ops		
ZIP	100 MB	$ 20	10^-8	400 ops		

Television Set

Nipkow rotating disk

Black & White NTSC

Color NTSC

Projection TV NTSC

CRT-->LCD/Plasma/DLP

Desktop Computer

HDTV (720p, 1080i))

HDTV (1080p)

- Vacuum Tube
- Transistor
- Integrated Circuit
- Specialized ICs

- Macrovision
- CSS
- HDCP

C-Team

- Broadcasting
- Analog Cable
- SATV
- Digital

Testing Issues

- Develop a test suite for each generation of TV
- Organize the tests into columns
- What pattern do you detect as the product evolves?

Product	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
	Size	Cost	Energy use	Image quality	Trouble shooting	Turn on time
BW	15 inch	500	200 W	sharpness	Tube tester	2 min
COLOR	30-inch	500	50 W	purity	PCB swap	1 min
Projection	50-inch	2000	150 W	convergence	Alignment	30 sec
LCD etc	50-inch	4000	100 W	stuck pixels	Digital	10 sec
HDTV	56-inch	2500	200W	compression	Digital	5 sec

Pick Other Examples

- Horse & buggy to the modern auto
- The airplane
- Earth orbiting satellites - Sputnik to Imaging
- Space exploration probes
- Computers from Eniac to the 4096-way SGI's
- Kitchen stove
- Home construction
- News delivery from town crier to Youtube