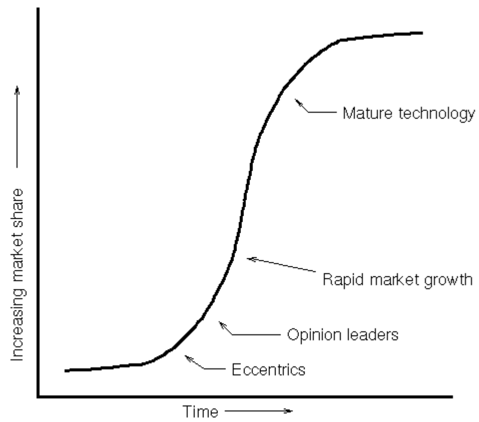
Technology Transfer Model
S-curve Growth

How do new technologies get adopted?:



Transition models

Gatekeeper -- Generally required. It is a member of infusing organization who is first sold on the new technology

Berniker model -- 4 methods for technology transfer:

- People mover model -- Personal contact between the developer and user (i.e., gatekeeper)
- Communication model -- Use of publications to broadcast information about new technology
- On-the-shelf model -- Use of a “parts catalog'” to advertise the new product.
- Vendor model -- Hire an organization to improve your technology

Zelkowitz addition to Berniker model

People Mover Model has three components:

- Spontaneous gatekeeper – Role assumed by organization member (Original gatekeeper)
- Assigned gatekeeper – Imposed by management to infuse new technology (Marginally successful)
- Umbrella gatekeeper – Imposed by another agency (e.g., government) (Generally not successful)

Umbrella gatekeeper least successful – Examples: Mandating Ada, NIST FIPS, GOSIP. On the other hand, C++, TCP/IP, Spreadsheets grew rapidly via the spontaneous gatekeeper.

Effective Transfer

- Spontaneous gatekeeper seems most effective in technology transfer.
- Problem: It is an importation process.
  - New organization must want to use the new technology.
  - It cannot be imposed successfully from the outside.

SEI MODEL

Fowler model from SEI -- A push--pull model:

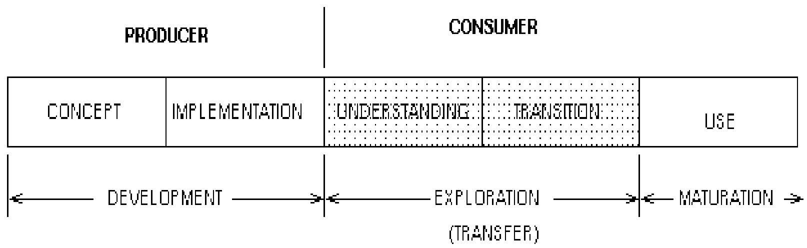
Producer → Advocate → Receptor → Consumer

Advocate and Receptor can be in either producer or consumer organization

Redwine-Riddle Study

- Technology maturation takes time:
- From Redwine – Riddle study (1985):
- Studied 17 software engineering technologies of the 1960s and 1970s (e.g., spreadsheets, UNIX)
- Required an average of 17 years from concept to maturation
- Required an average of 7.5 years after initial development to widespread availability in industry
- Similar times compared to other engineering disciplines

Technology Maturation Life Cycle (from Redwine and Riddle)



Technology Maturation Phases

Concept: Original concept introduced

Implementation: Initial implementation of technology

Exploration (understanding): Others experiment with technology, expand and modify it

Exploration (transition): Technology spreads across industry

Use: Mature when 70% of industry uses it

Technologies generally require 17-25 years to mature.

Corporate infusion of a new technology generally required 5-7.5 years

Software Engineering Technology Transfer

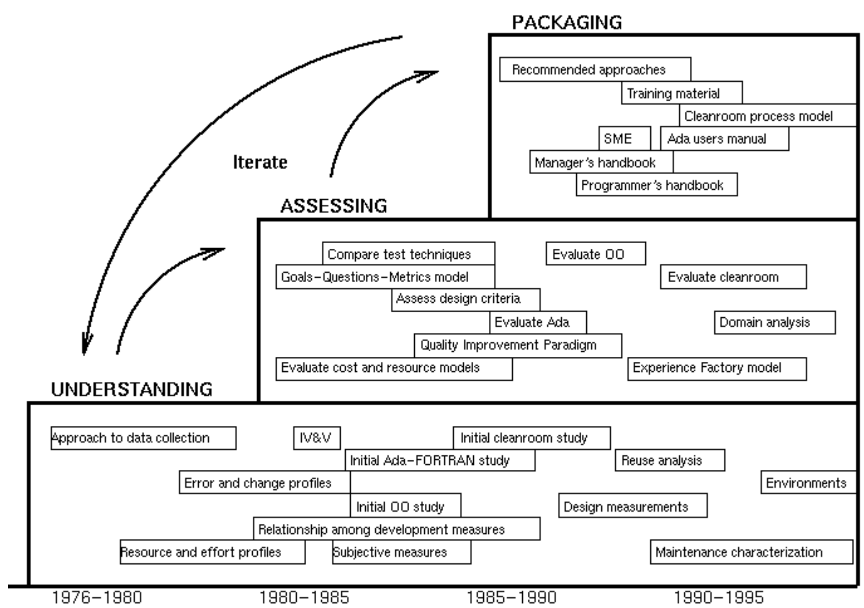
Experiences with the NASA/GSFC Software Engineering Laboratory

Background of the SEL:

- Began in 1976 to study software development
- Typical applications: attitude ground support software and dynamic simulators for onboard computers for unmanned spacecraft
- Characteristics:
  - Size from 10K to 500K source lines
    - FORTRAN dominant language; Some Ada; Now switching to C++
  - Typically 10-15 people for 18--24 months
  - Mixture of contractor and government personnel
  - Moving now to client--server architectures.
  - Over 125 projects; 500MB Oracle database

Many studies of effects of process changes on development in SEL environment

NASA Tech Transfer Process



NASA Tech Transfer

Fundamental issues for initial study:

- How does NASA think technology transfer takes place?
- How does technology transfer really take place?
- What is infusion process of transitioning a new technology

Software Engineering Technology Transfer

Technology Transfer is generally product oriented:

- In most engineering disciplines, the process is centered in the product.

Software engineering does not yet fulfill that model – Processes describing actions to take are as important as the tools that are used.

For example, many of the technologies explored by the GSFC Software Engineering Laboratory are procedures only and not tools:

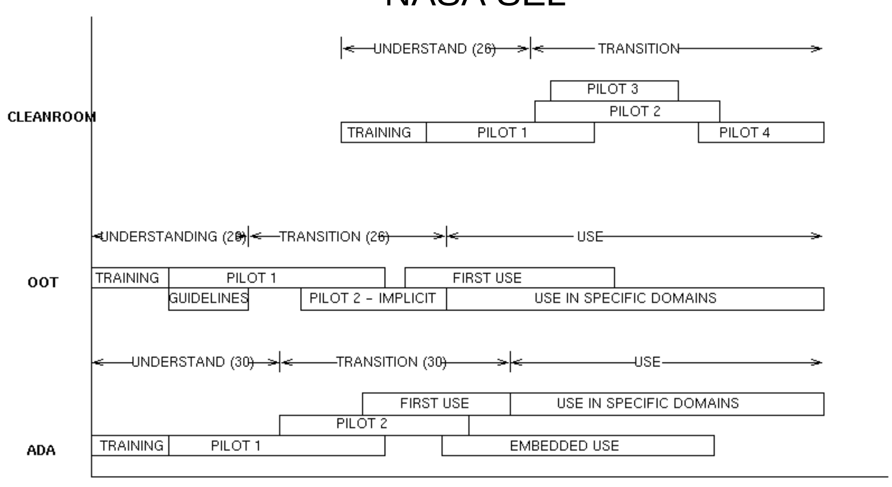
- Object oriented technology
- Goals/Question/Metrics model
- Measurement
- Cleanroom
- Inspections

Examples of transferred technologies

TOTAL REPLIES	**44**	**FROM NASA**	**12**
Workstations,pcs	**27**	**Object oriented**	**12**
Object oriented	**21**	**Networks**	**10**
GUIs	**17**	**Workstations,pcs**	**8**
Process models	**16**	**Process models**	**7**
Networks	**16**	**Measurement**	**5**
C and C++	**8**	**GUIs**	**4**
CASE tools	**8**	**Structured design**	**3**
Databases	**8**	**Databases**	**2**
Desktop publish	**8**	**Desktop publish**	**2**
Inspections	**7**	**Development meth**	**2**
Email	**7**	**Reuse**	**2**
Measurement	**6**	**Cost estimation**	**2**
		Comm. Software	**2**

Survey of software professionals – What 10 technologies (out of a list of over 100) have helped your productivity the most?

Example Technology Infusion NASA SEL



NASA Survey Conclusions

NASA mechanisms do not address software engineering technologies Well.

- Technology infusion is generally not addressed.
- Process technology is similarly not addressed.

Technology transfer is more than simply understanding the new technology.

- Time to understand technology is generally on the order of 2.5 years.
- Transition time at least as long as understanding time.

Validation in Software Engineering

Lots of technology development

Rapid change today within our technological society

But software failures are all too common

Why such failures?

Often there is a lack of validation before using a new technology

- **Anecdotal evidence that we don't validate our claims**
- **Study by Tichy (1995) that 50% of software engineering papers do not have validation;**
- **Only 15% in other scientific fields**

We need measurements (can't have a software engineering course without this comment):

"I often say that when you can measure what you are speaking about, and express it in numbers, you can know something about it. But when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind." – Lord Kelvin

Validation in Software Engineering

- But we also need relevant measurements:

"The government is very keen on amassing statistics – they collect them, add them, raise them to the nth power, take the cube root and prepare wonderful diagrams. But what you must never forget is that every one of those figures comes in the first instance from the village watchman, who just puts down what he damn pleases."

– British economist Josiah Stamp, 1929

Purpose of measurement?

"All we can ask of a theory is to predict the results of events that can be measured. This sounds like an obvious point, but forgetting it leads to the so-called paradoxes that popular writers of our culture are fond of exploiting." – Leon Lederman, Nobel Laureate physicist

What is science?

"Learn from science that you must doubt the experts. … Science is the belief in the ignorance of experts." – Richard Feynman, Nobel Laureate physicist

Experimental Models for Software Research

- But in Computer Science:
  – Our theories are our tools and techniques
  – All too often, we don't appreciate the “science” in our title
  – Validation, experimentation, and measurement seem to be lacking
- Recognition that we need to understand how to experiment in software engineering
- Problems:
  – Models mostly taken from social science domain.
  – View experimentation as the replication of a hypothesis under varying controlled conditions

Can we take larger view of experimentation that applies in the software domain?

What are experiments?

Different models:

Replicated experiments

- Chemistry – Rows of test tubes
- Psychology – Rows of freshmen students working on a task

Observations of what happens

- Medicine – Clinical trials
- Astronomy – Observe events if and when they occur

Data Mining of completed activities

- Archaeology – Dig up the past
- Forensic investigations – recreate what happened

- How do these relate to Software?
- What data does each method generate?

Basic Data Collection Models

Impact on the process being studied:

- Active methods – An effect on the process being studied
- Passive methods – No effect on process being studied

Based upon work of M. Zelkowitz and D. Wallace about 1995 at NIST

Classes of methods

- Controlled method – Multiple instances of an observation in order to provide for statistical validity of the results. (Usually an active method.)
- Observational method – Collect relevant data as it develops. In general, there is relatively little control over the development process. (Weakly active, although may be passive.)
- Historical method – Collect data from completed projects. (Passive methods.)

These three basic methods have been classified into 12 data collection models.

(We will also consider one theoretical validation method, yielding 13 validation methods)

Controlled methods

Replicated – Several projects are observed as they develop (e.g., in industry) in order to determine the effects of the independent variable. Due to the high costs of such experiments, they are extremely rare.

Synthetic environments – These represent replicated experiments in an artificial setting, e.g., often in a university.

Dynamic analysis – The project is replicated using real project data.

Simulation – The project is replicated using artificial project data.

The first 2 of these generally apply to process experiments while the last two generally apply to product experiments.

Observational methods

Project monitoring – Collect data on a project with no preconceived notion of what is to be studied.

Case study – Data collected as a project develops by individuals who are part of the development group. (Often used in SEL.)

Field Study – An outside group collects data on a development. (A weaker form of case study.)

Historical methods

Literature search – Review previously published papers in order to arrive at a conclusion. (e.g., Meta-analysis - combining results from separate related studies)

Legacy data – Data from a completed project is studied in order to determine results.

Lessons-learned data – Interviews with project personnel and a study of project documentation from a completed project can be used to determine qualitative results. (A weak form of legacy data.)

Static analysis – Artifacts of a completed project are processed to determine characteristics.

But List of Methods is Incomplete

Assertions: What do software engineers often do?

- For a new technology validation often consists of:
  "I tried it and I like it"

- Validation often consists of a few trivial examples of using the technology to show that it works.
- Added this validation as a weak form of case study under the "Observational Method:"

Assertion – A simple form of case study that does not meet rigorous scientific standards of experimentation.

Theoretical validation – A form of validation based upon mathematical proof.

Summary of validation methods

Summary: 13 methods
- 11 experimental methods
- assertion (weak experimental validation)
- theoretical validation

Evaluation of this classification

Review of 1995 Tichy study:
- Reviewed 403 papers
- Sources: ACM journals and conferences, IEEE TSE
- Classification of papers
  - Formal theory --Proofs
  - Design and modeling--Designs which are not formal
  - Empirical study--Evaluation of existing technology
  - Hypothesis testing--Experiments to test a hypothesis
  - Other--Anything else, e.g., surveys

Conclusions from Tichy study

Those relevant to current study:

- 40% of computer science papers without validation
- 50% of software engineering papers without validation
- Comparable numbers are neuroscience (12%) and optical engineering (15%)
- But only considered design and modeling papers.

Perhaps too narrow a view

NIST Evaluation

- Performed by Zelkowitz and Dolores Wallace
- New literature search: Papers from 1985, 1990, 1995
- Sources: 612 papers reviewed
  - IEEE Software --- a technical magazine
  - Transactions on Software Engineering - research journal
  - ICSE proceedings --- a conference

Can we detect changing trends over 10 years?

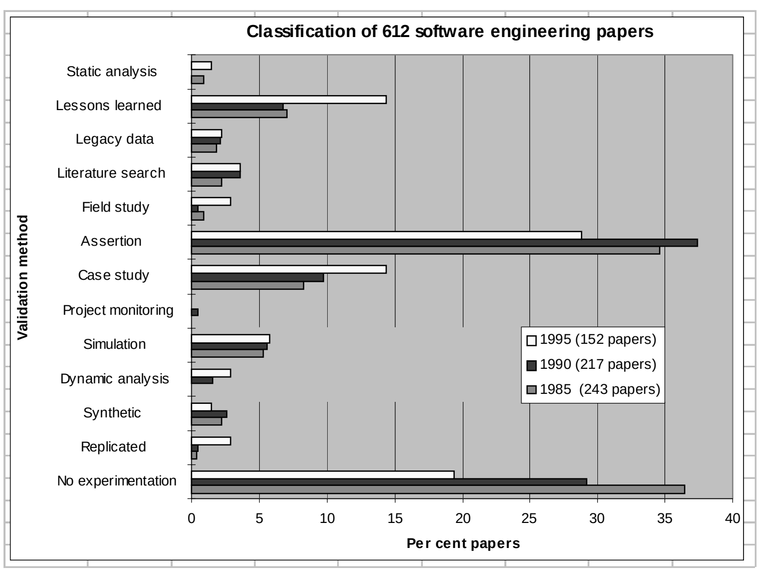
Added 2 more classifications to above 13:

- Not applicable --- The paper does not discuss a new technology, e.g., a survey paper.
- No experimentation --- The paper presents a new technology, but makes no claims as to experimental validity. These are the papers that SHOULD have validation of some form.

Summary of paper classifications

SUMMARY TOTALS	85			90			95			Ttl
Method	**ICSE**	**Soft**	**TSE**	**ICSE**	**Soft**	**TSE**	**ICSE**	**Soft**	**TSE**	
Not applicable	6	6	3	4	16	2	5	7	1	50
Theoretical	3	1	18	1	0	19	3	0	7	52
No experimentation	13	10	38	7	8	22	7	3	7	115
Replicated	1	0	0	0	0	1	1	0	3	6
Synthetic	3	1	1	0	1	4	0	0	2	12
Dynamic analysis	0	0	0	0	0	3	0	0	4	7
Simulation	2	0	10	0	0	11	1	1	6	31
Project monitoring	0	0	0	0	1	0	0	0	0	1
Case study	5	2	12	7	6	6	4	6	10	58
Assertion	12	13	54	12	19	42	4	14	22	192
Field study	1	0	1	0	0	1	1	1	2	7
Literature search	1	1	3	1	5	1	0	3	2	17
Legacy data	1	1	2	2	0	2	1	1	1	11
Lessons learned	7	5	4	1	4	8	5	7	8	49
Static analysis	1	0	1	0	0	0	0	0	2	4
Yearly totals	56	40	147	35	60	122	32	43	77	612

Summary graph



Quantitative Observations

- Most prevalent validation mechanisms were lessons learned and case studies, each about 10%
- Simulation was used in about 5% of the papers, while the remaining techniques were each used in under 3% of the papers
- About one-fifth of the papers had no experimental validation
- Assertions (a weak form of validation) were about one-third of the papers
- But percentages of no experimentation dropped from 26.8% in 1985 to 19.0% in 1990 to only 12.2% in 1995. (Perhaps a favorable trend?)

Qualitative Observations

- We were able to classify every paper according to our 13 categories, although somewhat subjective (e.g., assertion versus case study).
- Some papers can apply to 2 categories. We chose what we believed to be the major evaluation category.
- Authors often fail to clearly state what their paper is about. Its hard to classify the validation if one doesn't know what is being validated.
- Authors fail to state how they propose to validate their hypotheses.
- Terms (e.g., experiment, case study, controlled experiment, lessons learned) are used very informally.

MAJOR CAVEAT

The papers that appear in a publication are influenced by the editor of that publication or program committee. The editors and program committees from 1985, 1990, and 1995 were all different. This then imposes a confounding factor in our analysis process that may have affected our outcome.

Overall Observations

- Many papers have no experimental validation at all (about one-fifth), but fortunately, this number seems to be dropping.
- BUT too many papers use an informal (assertion) form of validation. Better experimental design needs to be developed and used.
- Lessons learned and case studies each are used about 10% of the time, the other techniques are used only a few percent at most.
- Terminology of how one experiments is sloppy. We hope a classification model, such as ours, can help to encourage more precision in the describing of empirical research.

Comparison to Other Fields

- We decided to look at several other disciplines for comparison, An informal study. No attempt at choosing the “best” journal in each field.
- Journals:
  - J 1 – Measurement Science and Technology, (Devices to perform measurements)
  - J 2 – American Journal of Physics, (Theory and application of new physical theories)
  - J 3 – Journal of Research of NIST, (Research on measurement and standardization issues)
  - J 4 – Management Science, (Queueing theory and scheduling problems)
  - J 5 – Behavior Therapy, (Clinical therapies)
  - J 6 – Journal of Anthropological Research, (Study of human cultures)

Summary of paper classifications

Method	J1 %	J2 %	J3 %	J4 %	J5 %	J6 %	TTL	%
NA		2	5			1	8	---
None	16	58	7	21	6	31	26	20
Replicated		5	4	4	12		5	
Synthetic			4	11	29		9	
Dynamic anal.	32	5	19	11			17	
Simulation			15	32			13	
Proj. Mon.								
Case study	40	16	41		6	8	26	
Assertion	8	4	11			8	7	5
Field study				4	18		4	
Liter. Search	4	11	7	7	24	23	14	
Legacy data					6	23	4	
Lessons learn		5				8	2	
Static anal.								
Paper count(#)	25	21	32	28	17	14	137	

Note clustering of techniques across journals
No attempt to summarize across fields, except for experimentation and assertions

Results from Other Fields

No experimentation plus assertion data much lower than in software engineering (25% versus 55%)

Each field has a characteristic data collection model:

- Physics --- dynamic analysis and simulation (repeated experiments)
- Psychology --- replicated and synthetic (repeated trials of individuals)
- Anthropology --- legacy data (historical data)

Literature search more accepted model for publication. (Does this refer to publication of similar studies that are frowned upon in computer science?)

In conclusion ...

- We have proposed a 13-way approach toward developing a quantitative model of software experimentation. It seems applicable to the software engineering literature.
- In a 1992 report from the National Research Council the Panel on Statistical Issues and Opportunities for Research in the Combination of Information recommended:

  "The panel urges that authors and journal editors attempt to raise the level of quantitative explicitness in the reporting of research findings, by publishing summaries of appropriate quantitative measures on which the research conclusions are based ..."
- In general, software engineering experimental validation is probably not as bad as folklore says, but could stand to do a better job.

WHY DOESN'T INDUSTRY "BUY" THIS?

Industry:

- Ignores results from archival journals
- Believes in unsubstantiated rumors

Research community:

- Doesn't require validation
- Doesn't perform validations as thorough as necessary

There is a "disconnect" between these 2 cultures

EXPERIMENTAL VALIDATION METHODS

Case study
Dynamic analysis
Field study
Lessons learned
Legacy data
Project monitoring
Literature search
Replicated experiment
Simulation
Static analysis
Synthetic study
Theoretical analysis

INDUSTRIAL METHODS

Additional methods often used by industry:

Expert opinion – use the opinion of experts. This can take the form of hired consultants brought in to teach a new technology or attendance of a trade show where various vendors demonstrate their products.

Edicts – changes required by an outside agent.

Feature analysis – a study of the features of the new technology and a subjective evaluation of its impact on the development process. Often used to compare two alternatives.

Compatibility studies – studies used to test whether various technologies can be combined or if they interfere with one another.

- *model problems* – narrowly defined problems that the technology can address.
- *demonstrator study* – scaled-up application development, with some attributes (e.g., performance, documentation) reduced in order to limit costs or development time.

Pilot study - This is a full-scale implementation using the new technology.

