

**LIVE-THRU CASE
HISTORIES:
OVERCOMING AVERSION
TO SOFTWARE PROCESS
THROUGH
CONTROLLED FAILURE**

**LAWRENCE BERNSTEIN AND
DAVID KLAPPHOLZ**

STEVENS INSTITUTE OF TECHNOLOGY

THE PROBLEM

25% -75% OF ALL SOFTWARE PROJECTS FAIL:

- FAIL TO MEET STAKEHOLDERS' FUNCTIONAL REQUIREMENTS**
- FAIL TO MEET LOAD REQUIREMENTS**
- FAIL TO PERFORM AS SPECIFIED BECAUSE OF INSUFFICIENT TESTING**
- ARE CANCELLED BEFORE COMPLETION BECAUSE OF INADEQUATE COST, PERSONNEL, AND SCHEDULE ESTIMATION.....**

THE REASON: INDUSTRY

**•LACK OF TRAINING IN /
ACCEPTANCE OF SOFTWARE
PROCESS (BEST PRACTICE) ON THE
PART OF SOFTWARE DEVELOPERS
AND SOFTWARE PROJECT
MANAGERS**

UNIVERSITY I: FACULTY

- **MOST CS FACULTY ARE CONCERNED WITH ADVANCING THE *STATE OF THE ART*, I.E., SOFTWARE TECHNOLOGY.**
- **MANY ARE EFFECTIVELY UNAWARE OF, AND ONLY MILDLY INTERESTED IN, THE *STATE OF THE PRACTICE*, I.E., THE (MEDIUM- AND LARGE-SCALE) SOFTWARE DEVELOPMENT PROCESS.**
- **MANY DEEM COURSES IN SOFTWARE ENGINEERING TO BE OF INTELLECTUALLY LOW CALIBER, AND OF LITTLE VALUE.**

Personally, I don't view
proselytizing for "best
practice" to be the
university's job

UNIVERSITY II: CS STUDENTS

- THE TYPICAL COMPUTER SCIENCE STUDENT:
 - IS TRAINED IN THE SOLUTION OF SMALL WELL-DEFINED PROBLEMS WHOSE SOLUTIONS PROVIDE IMMEDIATE FEEDBACK
 - HAS ABOVE-AVERAGE SKILL AT THIS TYPE OF ENDEAVOR; HAS, IN FACT, PROBABLY DECIDED TO MAJOR IN COMPUTER SCIENCE FOR THIS VERY REASON.
- HAS LESS THAN AVERAGE SKILL IN HANDLING:
 - ILL-DEFINED PROBLEMS
 - PROBLEMS WHOSE SOLUTIONS PROVIDE ONLY LONG-TERM PAYOFFS
 - PROBLEMS IN GROUP DYNAMICS.

WORSE YET

•FOLLOWING STATE OF THE PRACTICE (DISCIPLINE) IS NOT FUN

•STAYING UP LATE HACKING AND EATING PIZZA IS GREAT FUN

•MANY/MOST CS STUDENTS ARE AVERSE TO BEING TAUGHT STATE OF THE PRACTICE (SOFTWARE PROCESS).

... BECAUSE THEY DON'T KNOW THAT THEY ARE LIKELY TO FAIL WITHOUT IT

PROPOSED SOLUTION

USE CASE HISTORIES OF FAILED SOFTWARE PROJECTS AS A MEANS OF MOTIVATING STUDENTS TO A REALIZATION OF THE IMPORTANCE OF SOFTWARE PROCESS

BUT

- RECOUNTING FAILED CASE HISTORIES CAUSES STUDENTS ONLY TO RECOGNIZE THE STUPIDITY OF OTHERS

- HAVING STUDENTS READ FAILED CASE HISTORIES HAS THE SAME (NEGLIGIBLE) EFFECT

SO

CAUSE STUDENTS TO *LIVE THROUGH* FAILED CASE HISTORIES

LIVE-THRU CASE HISTORIES I

**•SELECT A SET OF SOFTWARE
PROCESS ISSUES TO BE MOTIVATED,**

E.G.:

**•NEED FOR RIGOROUS
REQUIREMENTS ENGINEERING**

**•NEED FOR UP-TO-DATE
DOCUMENTATION OF A PROJECT IN
PROGRESS**

**•NEED TO IDENTIFY RISKS AND TO
DEVELOP CONTINGENCY PLANS**

LIVE-THRU CASE HISTORIES II

- SELECT A CASE HISTORY IN WHICH THE PROJECT FAILED BECAUSE ONE OR MORE OF THE CHOSEN ISSUES WAS IGNORED OR EXECUTED BADLY**

AND

- EITHER GIVE STUDENTS THE SAME PROBLEM/ PROJECT AS WAS GIVEN TO THE ORIGINAL DEVELOPERS (4-5 WEEKS)**

- OR GIVE STUDENTS THE JOB OF JUST DESIGNING THE SOFTWARE, COMING UP WITH A LIFE-CYCLE PLAN, ETC. (2-7 DAYS)**

EXAMPLE: OVERDUE BOOK-NOTICE SYSTEM FOR PUBLIC SCHOOLS

PROBLEM

AN ELEMENTARY SCHOOL OF 500 STUDENTS IS USING A MANUAL METHOD FOR TRACKING BOOKS ON LOAN. WHEN A BOOK IS TAKEN FROM THE LIBRARY THE CARD IS TAKEN FROM THE BACK-OF-BOOK POCKET AND FILED BY DATE. A BOOK MAY BE BORROWED FOR TWO WEEKS. WHEN A BOOK IS RETURNED THE CARD IS PUT BACK INTO THE BACK-OF-BOOK POCKET.

A BOOK THAT IS NOT RETURNED IN TWO WEEKS IS CONSIDERED OVERDUE AND AN OVERDUE NOTICE IS SENT TO THE BORROWER. A STUDENT WHO DOES NOT RETURN A BOOK WITHIN THREE WEEKS IS GIVEN A SECOND NOTICE. A BOOK NOT RETURNED IN FOUR WEEKS GETS A THIRD NOTICE AND IS REPORTED ON A 'CRITICAL OVERDUE' LIBRARIAN'S CLERKS WRITE OVERDUE BOOK NOTICES ON ONE HALF OF A SHEET OF PAPER FOR ABOUT 200 BOOKS EACH WEEK. THE NOTICES ARE DISTRIBUTED WEEKLY.

**THIS LAYOUT OF THE OVERDUE BOOK NOTICE
WAS USED FOR MANY YEARS:**

GLENWOOD SCHOOL OVERDUE BOOK NOTICE

BOOK TITLE:

STUDENT NAME:

TEACHER NAME:

DATE OF NOTICE:

NOTICE 1 2 3 (CIRCLE ONE)

THIS BOOK IS OVERDUE PLEASE RETURN IT PROMPTLY

PROJECT

AUTOMATE THE OVERDUE BOOK NOTICE PROCESS USING A COMPUTER BORROWED FROM THE COMPUTER CLASS. THIS IS THE FIRST COMPUTER AUTOMATION PROJECT IN THE SCHOOL. DATA MAY NOT BE LEFT ON THE COMPUTER FROM WEEK TO WEEK. THE COMPUTER IS NOT NETWORKED.

ORIGINAL DEVELOPMENT

SYSTEM WAS REJECTED BECAUSE:

•*OVERDUE NOTICES MUST BE SORTED, FIRST BY TEACHER NAME, THEN, FOR EACH TEACHER BY CLASS: OTHERWISE THE LIBRARIAN MUST SORT THEM BY HAND*

•*DATA ENTRY MUST BE BY BOOKS NOT RETURNED, RATHER THAN BY BOOKS RETURNED: OTHERWISE ENTRY TAKES TOO LONG*

ADDITIONAL GLITCH

**AT THE MIDPOINT OF THE PROJECT
EACH TEAM'S BEST DEVELOPER AND
ANOTHER, RANDOMLY CHOSEN, TEAM
MEMBER WERE REMOVED FROM THE
TEAM AND RE-ASSIGNED TO A
DIFFERENT TEAM.**

**•LIVE-THRU DEVELOPMENT
PROJECT (STEVENS: 2000, 2001,
2002)**

**•LIVE-THRU HOMEWORK
(MONMOUTH U: 2002; USC 2002)**

**•LIVE-THRU EXAM (STEVENS:
2000, 2001, 2002; USC: 2002)**

**AT USC WE FOUND THAT DOING
A VERY SIMPLE LIVE-THRU
HOMEWORK WAS MORE
EFFECTIVE IN TEACHING THE
NOTION OF “MODEL CLASHES”
THAN READING ABOUT A MUCH
LARGER FAILED PROJECT**

PROJECT

Automate the overdue book notice process using a computer in the computer class' room. This is the first automation project in this rural school. Data may not be left on the computer from week to week. The computer is not networked.

You need not actually implement the system. Rather:

1. Design the screens that the librarian will use
2. Write a 1-2 page user's manual for the system
3. Do a flowchart of the system
4. Design and list a reasonable number of (test case) inputs to the system and show what the output would be if the test cases were input in the listed order