

CS 350, slide set 7

M. Overstreet
Old Dominion University
Spring 2005

Reading

- Team Software Process text, Ch. 1, 2, 3

Topics

- Intro to TSPI
- What's coming in the rest of the semester?
- Some problems and warnings

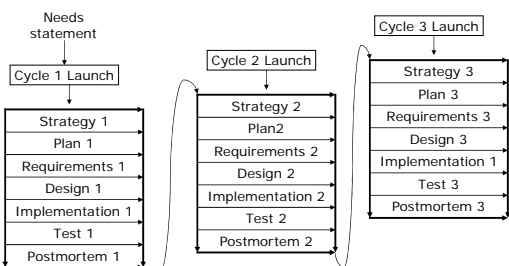
Quandary

- Most of the technology you will need to understand to be successful in your jobs doesn't exist yet.
- Employers identify problem solving as the key employee skill.
- In some crucial ways, the main thing to learn is a *process* for dealing with new problems.

TSPi overview

- i stands for instruction.
 - Subset of TSP
- Focus:
 - Based on PSP
 - Scripts, measurements, metrics
- Teams & roles
 - Different members responsible for different parts of joint project
- Develop complete product in several complete cycles

TSPi Structure and flow



TSPi Development Script - 1

Purpose	Guide team through dev. software project
Entry Criteria	<ul style="list-style-type: none"> ■ Instructor to guide and support project ■ Students know PSP ■ Instructor has project description ■ Instructor has described project objectives
Exit Criteria	<ul style="list-style-type: none"> ■ Completed project ■ Completed user documentation ■ Completed and current project notebook ■ Documented team evaluations and cycle reports

TSPi Development Script - 2

Wk	Step	Activities
1	Review	<ul style="list-style-type: none"> ■ Read TSP ch. 1 and 2.
2	LAU1	<ul style="list-style-type: none"> ■ Assign teams and roles. ■ Read TSP ch. 3, App B and one of ch. 11-15.
	STRAT1	<ul style="list-style-type: none"> ■ Produce conceptual design, establish dev. strategy, make size estimates and assess risk. ■ Read TSP ch. 4.
3	PLAN1	<ul style="list-style-type: none"> ■ Produce cycle 1 team and engineer plans ■ Read TSP ch. 5 & App C.
4	REQ1	<ul style="list-style-type: none"> ■ Define and inspect cycle 1 requirements. ■ Produce system test plan and support materials. ■ Read TSP ch. 6 and test sections of ch. 9.
4	DES1	<ul style="list-style-type: none"> ■ Produce and inspect cycle 1 high-level design. ■ Produce integration test plan and support materials. ■ Read TSP ch. 7.

TSPi Development Script - 3

Wk	Step	Activities
5	IMP1	<ul style="list-style-type: none"> ■ Implement and inspect cycle 1. ■ Produce unit test plan and support materials. ■ Read TSP ch. 8.
6	TEST1	<ul style="list-style-type: none"> ■ Build, integrate, and system test cycle 1. ■ Produce user documentation for cycle 1. ■ Read TSP ch. 9.
7	PM1	<ul style="list-style-type: none"> ■ Conduct a postmortem and write cycle 1 final report. ■ Produce role and team evaluations for cycle 1. ■ Read TSP ch 10, 16, 17, and 18.
	CYCLE 2	<ul style="list-style-type: none"> ■ Repeat above for cycle 2 (we won't have time for this).
	CYCLE 3	<ul style="list-style-type: none"> ■ Repeat above for cycle 3 (we won't have time for this).

Why projects fail

- Rarely for technical reasons
 - Internal politics
 - Team does not bind
 - Fail to develop rapport with customers
 - People will fight over meaningless issues
- Pressure is a problem
 - Having a plan of action helps
 - Know real issues that must be resolved rather than worrying about imaginary problems

Common team problems

- Ineffective leadership
 - Few people are natural leaders, but can get better with practice
 - Beneficial to have effective examples (people)
- Some people don't know how to compromise
- Lack of participation
- Procrastination/lack of confidence
- Poor quality
- Function creep
- Poor peer evaluation

Team definition

- For TSPi, a team consists of
 - at least 2 people (TSP designed for 5), who
 - are working toward a common goal, where
 - each member is assigned specific responsibilities and where
 - successful completion of project requires team members to contribute.

Jelled teams

- Whole greater than sum of parts
- Great satisfaction for members
- Necessary conditions
 - Task to be performed clear
 - Team responsible clearly identified
 - Including who is and is not on team
 - Team has control over tasks
 - Can be dangerous to team members
 - Can't "not do it" attitude
 - Hard on personal relationships (spouses, significant others)
 - See "Soul of a new machine" by Tracy Kidder
 - Identified as one of best 100 books of 20th century

How to build teams

- Common goals
- Assigned roles
 - Most people want to contribute.
 - Each person needs specific task to complete that he/she understands, and
 - Peer pressure has an effect.
- Need plans
 - Strategy for achieving goals
- Communication
 - Weekly meetings – if possible part of recitation time

Problems & warnings

- TSP instruction problem:
 - Students learn TSP by doing a "big" project
 - Students need to know TSP before they start
- So, we need to finish the TSP book by Tuesday so you can start the semester-long (well, half semester) project
 - And we can't
- You are the victims of an experiment!
 - Struggle with better ways of teaching what you need to know OVER THE LONG RUN
 - Changing views on how to do this

Launching a new team

- Defining goals for team and team members
- Defining roles
 - How the group is to be organized
 - Establish responsibilities of each role
 - Just makes it easier and quicker to divide up work
 - Still, everybody develops and tests code, everybody manages some aspect of the project
- Assigning roles

Goal considerations

- Aggressive but realistic
 - Here, we want to stretch your abilities, but not crush you
 - Avoid timid, safe goals
 - Should strive to achieve, but cannot be punished severely if not achieved
 - They matter (but they don't)

Identifying team goals

- Write them down
- Decide how to measure
- Explain why you picked them
- Give copy to other team members and to instructor
- Have the support manager put a copy in the project notebook

General comments on goals

- Should relate to how a user will perceive the product:
 - Quality
 - Utility
 - Costs
 - When available
- In 350, instructor and grader are the customers

Possible goals

- Attempt 1:
 - Produce a quality product
 - Run a well-managed project
 - Extend project beyond minimal
- These may seem too vague, but if concrete measurements are added:

Goals and metrics - 1

- Team goal 1: Produce quality product
 - Percent of defects found before 1st compile: 80%
 - Number of defects found in system test: 0
 - Requirements functions included at project completion: 100%

Goals and metrics - 2

- Team goal 2: Run a well-managed project
 - Error in estimated product size: < 20%
 - Error in estimated development hours: < 20%
 - Per cent of data recorded and entered in project notebook: 100%
 - Number of days project completed before deadline: 3

Goals and metrics - 3

- Team goal 3: extend project beyond minimal requirements (examples only):
 - Add ability to randomly generate inputs within range for at least 3 sensors (to support testing)
 - Develop harness to run 2 versions of project, feed them random sensor data, and report differences in output.

TSPi team members - 1

- Team leader
 - Resolves issues among members
 - Facilitates meetings
 - Much more, see text. See fig 3.1
 - e.g. decides how the project notebook will be kept
 - see appendix G for notebook specs and standards
- Development manager
 - Lead all development work
 - Much more, see text
- Planning manager
 - Lead team planning and progress tracking
 - Much more, see text

TSPi team members - 2

- Quality/Process manager
 - Lead quality planning and tracking
 - Act as inspection moderator
 - Much more, see text
- Support manager
 - Obtain needed support tools
 - Handle configuration management
 - Much more, see text

Team member goals

- Examples
 - Be a cooperative and effective team member
 - Average PEER eval. for helpfulness and support > 3
 - Average PEER eval. for overall contribution > 3
 - Produce quality products
 - Defect density at compile < 10/KLOC
 - Defect density at test < 5/KLOC
 - Defects found after unit test: 0

Example role goals

- Planning manager goal:
 - Accurately report team status every week to instructor
- Support manager goal:
 - No unauthorized changes made to baselined product
- Quality/Process manager goal:
 - All team inspections are properly moderated and reported

Launch script

Purpose	<ul style="list-style-type: none"> ■ To start teams on a development cycle ■ Students have read ch. 1, 2, 3, and reviewed NASA requirements
General	<p>The instructor describes TSPI objectives</p> <ul style="list-style-type: none"> ■ Form teams and assign team roles ■ Explain objective for the software ■ Establish meeting and reporting times <p>Steps 1, 2, and 3 are completed during the first meeting Steps 4 through 8 are completed during the second meeting</p>
Exit criteria	<ul style="list-style-type: none"> ■ Each student has completed and submitted INFO form ■ The development teams are formed and roles assigned. ■ The instructor has described the overall product objectives ■ The instructor has reviews and discussed the TSPI and team's role goals ■ Each team has agreed on goals, weekly meeting times, and the weekly data to report

Student information sheets - 1

Mail to cmo by next Monday, March 21

Rank from 1 (least) to 5 (most) your preferences for serving the the following roles:					
Team Leader	1	2	3	4	5
Development Manager	1	2	3	4	5
Planning Manager	1	2	3	4	5
Quality/Process Manager	1	2	3	4	5
Support Manager	1	2	3	4	5

Weekly meeting script

Purpose	To guide teams in conduction weekly meetings
Entry criteria	<ul style="list-style-type: none"> ■ All team members present ■ All team member have provided updated TASK, SCHEUDLE, and WEEK forms to the planning manager ■ The team leader has issued a meeting agenda
General	<p>In advance of the meeting, the team leader has:</p> <ul style="list-style-type: none"> ■ Asked team members for meeting agenda topics ■ Prepared and distributed the meeting agenda <p>The team leader leads the weekly meeting</p> <ul style="list-style-type: none"> ■ The quality/process manager records the meeting topics ■ Each team member generally reports his/her role work and development work at the same time ■ After the meeting, the team leader distributes the meeting report ■ Puts a report copy in the project notebook
Exit criteria	<ul style="list-style-type: none"> ■ The meeting report completed and placed in the project notebook ■ Updated team and programmer TASK, SCHEDULE, WEEK, and CSR forms in the project notebook ■ Updated copy of the ITL (issue tracking log) in the project notebook

Weekly forms

- See text (table 3.5) for meeting step by step details
 - Agenda review
 - Role reports
 - Engineer reports
 - Close meeting
- See text (table 3.7) for individual report instructions.

Term project - 1

- Goal: build several GCS systems so that their correctness can be compared.
- Problems:
 - Program large; perhaps not enough time
 - Insufficient mathematics background (based of prereqs for CS 350)

Term project - 2

- Approach:
 - Review requirements spec to identify pitfalls
 - Identify subset of requirements to implement
 - Only check subset of outputs

Term project - 3

- Each version will be a function which is called by a driver
- The driver must set up the execution environment for the GSC module
 - What's need in the driver? Check spec.
- Idea is that each group provides a driver to check their version, but that the version, without recompilation, can be used with an n-version driver.
- Since versions must be interchangeable, all must make the same assumptions about the execution environment.

Driver requirements?

- Does it call GCS once or repeatedly?
- How does it assign values to variables?
 - Nice if this does not require lots of work for tester
 - Comment: since this is directed at an n-version approach, abandon preparation of "correct" inputs?
- What does it output? And in what format?
 - Nice if easily readable by people
 - Nice if easily processed by more code
