

CS 350, slide set 8

M. Overstreet
Old Dominion University
Spring 2005

Reading

- TSP text, Ch. 4, 5
- Chapters 11 – 15, depending on your role.

Announcements - 1

- Group meetings in recitation
- Used to meet, set group goals, set up project notebook
- Agenda:
 - Be prepared to describe your background to group
 - Be prepared to state your goals for group project

Announcements - 2

- Code, etc. due tonight
- Forms due Wed.
- Role prefs. due tonight
- Due next Mon. night:
 - Team goals (approved by team)
 - May require e-mail, IM, phone discussions
 - Individual goals
 - Role goals
- Remember: let me know quickly if you have trouble contacting any team member!

Outline

- More forms
- Development strategy
- Development plan

A development strategy

- Lots of different approaches to software development
 - Develop entire system
 - Develop system in small pieces
- Most prefer cycle (but not always possible):
 - Code a little
 - Test a little
 - Have something running early-on
 - Always be ready to run demo (if feasible)
- Focus on uncertainty early
 - Identify and resolve risks

Steps – Planning and schedule

- Plan first
 - Big picture first, then refine details
- Can do top-level planning
 - We're lucky: we have real deadlines
 - Hard constraints eliminate possibly confusing choices
 - Review outline from last slide set.
 - Be ready to discuss with group on Friday
 - Feasible
 - Does it need changing?

Steps – Conceptual design

- Must omit details
- Identify major pieces
 - What are they
 - How will they likely interact
 - What depends on what?
 - Can some components be completed first and others brought in later
 - (Strategy: make sure you have something for the grader to run, even if some major parts are missing! -- if worst comes to worst)

Identify major risks

- Pieces you can't code?
- How to test?
- What are problems common to working in teams?
 - Some team members will likely have experience
- Try to anticipate so you can mitigate
 - Key components have two developers: coder and tester
 - Either could step in for the other if necessary

Productivity comes from reuse (but from your **own** code!)

- Identify code you already own that might reduce effort in project
- Plan reuse library
 - During design
 - Look for reuse from this class
 - Identify new components that have potential for reuse.

Strategy scripts: overview

Purpose	To guide team through a TSPI development strategy; have preliminary size and time estimates
Entry criteria	Everyone's read ch. 4 TSPI process discussed in class Objectives discussed in class Teams assigned and roles assigned Teams have agreed on goals for team and their own work
General	Dev strategy specifies order of tasks (code, test, etc) How to divide work within group It may change; keep it up to date Time estimates provide basis for work allocation
Exit criteria	Complete and documented dev. strategy Complete size & time estimates Complete configuration management plan Risks identified and entered in ITL log Conceptual design and completed STRAT form Updated project notebooks

Development Strategy - steps

1. Strategy overview
2. Establish strategy criteria
3. Produce conceptual design
4. Select development strategy
5. Produce prelim. estimates
6. Produce prelim. schedules
7. Assess risks
8. Document strat.
9. Produce config. mgmt plan

Possible development strategy ideas:

- Form alliance with another group to exchange test data.
 - No copying of data, but pooling might be ok if planned and discussed with instructor
- Form alliance with another group to exchange module object code
 - Rule: no exchange of source but look for other ideas
 - If anything like this is done; identify authors, etc.
- These will take time; are they worth it?
- Other ways to reduce work/improve grades?
- Note: intergroup activities are encouraged but be careful: plan, then check with instructor.

Development plan - 1

- One size does not fit all!
 - Simple tasks only require simple plans
 - Failure costs effect how much planning is appropriate
- Tight schedules effect appropriate planning detail
- Essence of teamwork in meeting commitments
 - Need to know what you're being asked to commit so you can determine if you can finish by the deadline
 - Need to balance work among team members

Development plan - 2

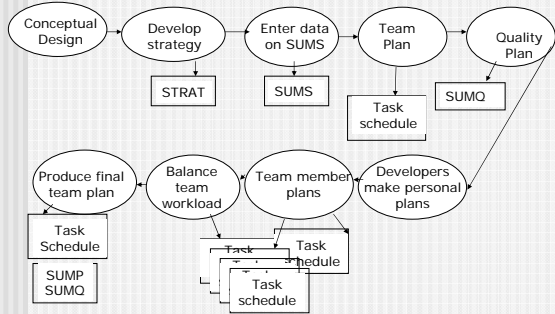
- Need to track progress precisely
 - Need to know that when x% of money has been spent that the project is x% complete (or when x% of schedule has been used up, x% of project is complete)
- Hence planned value and earned value tracking
 - Planned value of a task is its percentage of total project effort
 - Earned value of a task is 0 until it is 100% complete (to keep us from deceiving ourselves)

Planned-value example

Task	Plan hours	Cumulative hours	Plan Value	Cumulative Plan Value
Launch	29.5	29.5	8.0	8.0
Planning	27.5	57.0	7.5	15.5
System test plan	17.0	74.0	4.6	20.2
...
Documentation	33.0	354.5	9.0	96.7
Postmortem	12.2	367.0	3.3	100.0
Total	367.0		100.0	

Remember, these are team numbers, so divide by 5 to get average individual numbers!

TSPi Planning Overview



STRAT form

- List, item by item
 - Exactly what you plan to implement
 - Page and paragraph reference to Requirements Doc (from NASA)
 - (If the semester were longer) in what phase you plan to implement it. In 350, the planned order of development

Size summary (SUMS) form

- For each software unit, report estimates for:
 - size
 - base (if you are modifying existing code, this is the size of it)
 - deleted (code removed from base, if any)
 - modified (code modified in base)
 - code added
 - code reused
 - new + modified (this is the effort measure)
 - total in new unit
 - loc of new code that might be reused
- Repeat with actuals after unit is complete

TASK form

- For each task, include (one line/task):
 - planned time, by group member:
 - Team leader
 - Development mgr
 - Planning mgr
 - Quality/process mgr
 - Support manager
 - planned size/value
 - size
 - date
 - Planned value
 - actuals
 - hours
 - date

Sample SCHEDULE form

Week No	Date	Plan			Actual			
		Direct Hrs	Cum Hrs	Cum Planned Value	Team Hrs	Cum Hrs	Week Earned Value	Cum Earned Value
1	10/29	26.8	26.8	18.5	27.0	27.0	20.5	20.5
2	11/5	56.5	83.3	34.0	44.4	71.4	62.5	83.0
3	11/12	47.0	130.3	45.0	38.5	107.2	37.5	120.5
4	11/19							
5								
6								

Form WEEK – for weekly reports

Group Weekly Data		Planned		Actual	
Project hours this wk					
Project hours to date					
Earned value to date					
Total hours for tasks completed to date					
Team member weekly data		Hours	Hours	Planned	Earned
		Planned	Actual	Value	Value
Team leader					
Development manager					
Planning manager					
Quality/Process manager					
Support manager					
Development Tasks Completed		Hours	Hours	Earned	Planned
		Planned	Actual	Value	Week
Issue/Risk Tracking					
Issue/Risk name		Status			
Other Significant Items					

SUMQ form (Quality plan) -1

Summary Rates		Plan	Actual
LOC/hr			
% Reuse (of total LOC)			
% New Reuse (of new and changed LOC)			
Percent Defect-free (PDF)			
In compile			
In unit test			
In build and integration			
In system test			
Defect/page			
Requirements inspection		} Driver related	
HLD inspection			
Defects/LOC			
DLD review		} Pseudocode for each module?	
DLD inspection			
Code review			
Compile			
Code inspection			
Unit test			
Build and integration			
System test			

SUMQ form (Quality plan) -2

Defect ratios		Planned	Actual
Code review/compile			
DLD review/Unit test			
Development time ratios			
Requirements inspection/Requirements			
HLD inspection/HLD			
DLD/code			
DLD review/DLD			
Code review/code			
A/FR			
Review rates			
DLD lines/hour			
Code LOC/hour			
Inspection rates			
Requirements pages/hour			
HLD pages/hour			
DLD lines/hour			
Code LOC/hour			

SUMQ form (Quality plan) -3

Defect-injection rates (Defects/hr.)	Plan	Actual
Requirements		
HLD		
DLD		
Code		
Compile		
Unit test		
Build and integration		
System test		
Defect-removal rates (Defects/hr.)		
Requirements inspection		
HLD inspection		
DLD review		
DLD inspection		
Code review		
Compile		
Code inspection		
Unit test		
Build and integration		
System test		

SUMQ form (Quality plan) -4

Phase yields	Plan	Actual
Requirements inspection		
HLD inspection		
DLD review		
Test development		
Test inspection (added)		
DLD inspection		
Code review		
Compile		
Code inspection		
Unit test		
Build and integration		
System test		
Process yields		
% before compile		
% before unit test		
% before build and integration		
% before system test		
% before system delivery		

Plan Summary (SUMP)

Product size	Plan	Actual	
Requirements pages			
Other text pages			
High-level design			
Detailed design lines			
Base LOC (B) measured			
Deleted LOC (D)			
Modified LOC (M)			
Added LOC (A)			
Reused LOC (N)			
Total New and Changed LOC (N)			
Total LOC (T)			
Total New Reused LOC			
Estimated Object LOC (E)			
Upper Prediction Interval (70%)			
Lower Prediction Interval (70%)			
Time in Phase (hours)	Plan	Actual	Actual %
-- same steps as the SUMQ form +			
Postmortem			
Defects injected			
-- same steps as the SUMQ form			
Defects removed			
-- same steps as the SUMQ form			

TSPi Development plan script

Purpose	To guide a team through producing individual and team task, schedule, and quality plans
Entry criteria	<ul style="list-style-type: none"> ■ Team has development strategy and conceptual design ■ Members have read text Ch. 5
General	<p>The plan defines</p> <ul style="list-style-type: none"> ■ Time required to perform each task ■ Planned value for each task <p>The schedule plan gives</p> <ul style="list-style-type: none"> ■ Each person's planned time for each project week ■ Anticipated completion week for each task ■ Planned value for each week <p>This will likely involve some iterations</p>
Exit criteria	<p>Completed team and individual TASK and SCHEDULE forms</p> <ul style="list-style-type: none"> ■ Reviewed and approved complete task list <p>Completed SUMP, SUMO, and SUMS forms</p> <p>Update project notebook</p>

Plan steps - 1

Step	Activities	Description
1	Planning overview	How plan is produced How quality plan is produced
2	Sizes estimates in STRAT	Starting with conceptual design, Planning Manager leads team in identifying missing parts and their sizes; record on STRAT form
3	Produce task plan	Planning manager leads team through producing task list with team and individual time estimates; enter in TASK form
4	Produce schedule plan	Planning manager obtains estimated hrs each member plans to spend on project each week and enters this in SCHEDULE form; produces team TASK and SCHEDULE forms; reworks if needed

Plan steps - 2

5	Produce quality plan	<p>Quality/process manager leads the team through:</p> <ul style="list-style-type: none"> ■ reviewing team's quality objectives ■ estimated defects injected and defect-removal yields ■ generating and assessing trial SUMP and SUMO plans ■ Making need process adjustments to get a sat. plan
6	Produce individuals' plans	<p>Planning manager helps individual make personal plans:</p> <ul style="list-style-type: none"> ■ allocate tasks among team members ■ estimate time needed for each task ■ entering data in TASK and SCHEDULE forms ■ producing planned-value schedule & task comp. times
7	Balance team workload	<p>Planning manager leads team through:</p> <ul style="list-style-type: none"> ■ identifying workload imbalances ■ reallocating task to shorten schedule ■ producing balanced individual plans ■ producing the consolidation team plan (TASK, SCHEDULE, SUMP, SUMO)

Example forms

- Read chapter 5
- Contains worked-out examples of all of these forms
 - Some forms I only outlined just to talk about terms used on the forms
- Also contains justifications for several of the steps
- It should help you in setting goals for yield, defect removal rates, etc.

Conceptual design

- Tell me exactly what you plan to implement
 - Include object names, but no implementation details
 - Give around 1 sentence description of what each object does
- Goal:
 - High level view of how system will function
 - What pieces need to be implemented; how they're related
- Format: Text, pictures:
 - Whatever is easy to produce
 - (you will need to keep it current)
 - Whatever communicates effectively
 - Within team
 - To instructor & grader

Products

- List all of the products you will produce
 - Names
 - Brief description
 - Size estimates
 - Time estimates
- Include:
 - Conceptual design
 - Test plans
 - Test data
 - Tasks
 - Stubbed routines (if needed)
 - Drivers (if needed)

Tracking the work

- Goal: track progress and quality
- TSPi tool can help with this
- Steps (examples)
 - Record time in time log
 - Enter week tasks are completed
 - TSPi tool can generate TASK and SCHEDULE data
 - Enter defect in defect recording log
 - Enter size of component
 - TSPi tool can generate SUMP form
 - TSPi tool can generate SUMQ form
 - Submit to planning manager for WEEK rpt

Reuse comments

- Try to reuse code developed thus far in this class
- Team must decide which version to use
 - Team can compare members' versions and pick one
- If you can reuse any team member's code from other classes, do so.
 - Be sure you document this: in plans and in code
 - If you find other LEGAL code sources (say, on net), let me know
