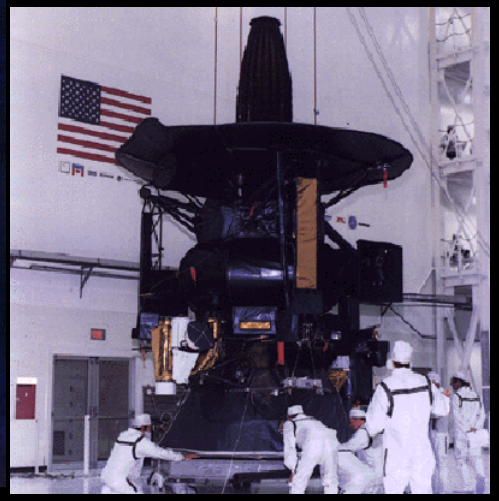
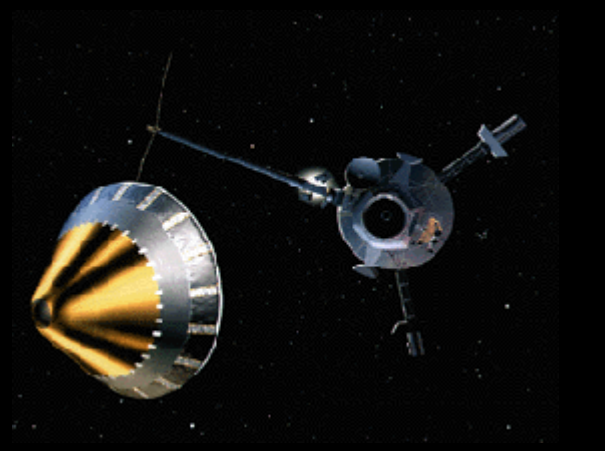
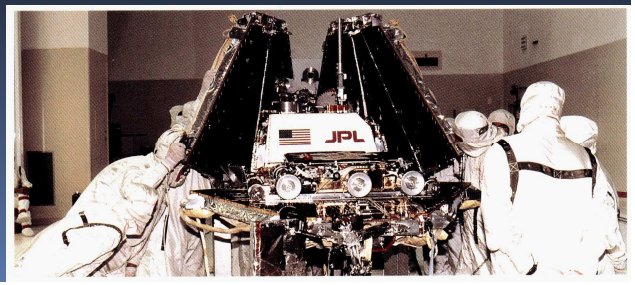




MANAGING FOR SOFTWARE RELIABILITY

Steven A. Stolper, Software Consultant
info@stolperconsult.com



Software Management affects Reliability

- Risk Management
- Software Estimation
- Development Practices*
- Dealing with the "Death March" (Crises)

Risk Management (Identification)

- Identify technical risk (mindfulness)
- Question until you understand
- Formulate strategy to mitigate
- Manage as a resource
- Examples:
  - New technology, New flight hardware, Remote engineering, Small staff, hardware arrives late in development, new development tools…
- Justify technology use

Risk Management Continued (Mitigation)

- Conjoint measurement:

XYZ PROJECT
Risk Assessment and Impact Model
November 3, 2009

Risk Threshold = 9

Expectation: 1 (very unlikely), 3 (50% likelihood), 5 (extremely high likelihood) Impact: 1 (lowest impact), 3 (medium impact), 5 (highest impact)

Risk area	Potential risk	Expectation of the risk (1 to 5)	Impact of the risk (1 to 5)	Severity of the risk (Expectation X Impact)	Mitigating strategies & controls	Decision summary	Owner
Schedule / Budget	Parts arrive late from suppliers	3	3	9	Safety margin of 10% incorporated into lead time for critical components	Specifics of how we mitigate this risk go here	J. Smith
Schedule / Design / Technical	Staff turnover	3	4	12	Technical leads will maintain knowledge of team member's work. Each technical lead will have a deputy who has "high-level knowledge" and "history"	Specifics of how we mitigate this risk go here	A. Thomas
Technical	First use of RTOS and compiler tool-chain	3	4	12	Hire at least one developer with previous experience with toolchain. Contract with vendor for developers to attend classes. Perform benchmark testing.	Specifics of how we mitigate this risk go here	A. Thomas

Technical Debt (Unintentional)

- Insufficient up-front design (architecture)
- Not enforcing good implementation practices
  - Modularity
  - Sanity checks
  - Comments / documentation
- Not enforcing good development practices
  - Concise specifications / requirements
  - Design reviews
  - Code Reviews
  - Monitoring check-ins
  - Fixing compiler warnings
- Major headaches in QA/ATLO or IN FLIGHT!
- A MESS IS NOT TECHICAL DEBT!!!

Why is Good Software Estimation Important?

- Avoid the "Death March"
  - Twelve-hour workdays and weekends
    - Mental fatigue
    - Dumb mistakes
    - Shortcuts
- Increases technical debt

Software Estimation

- Scheduling:
  - Work Breakdown Structure (WBS)
    - Activities required to complete project
  - Estimate
    - Work required to complete activities
  - Schedule
    - Ordering of activities and assignment of resources

Software Estimation Basics

- A target is not an estimate!
- What is a good estimate?
  - For commercial software, within 25% of actual, 75% of the time (Conte, Dunsmore, Shen)
  - Some organizations have achieved accuracy within 10%, but don't hold your breath.
  - Are you 90% confident? (best->worst)

The Cone of Uncertainty

Please see renditions of the "Cone of Uncertainty" widely available on the Internet

Estimating (cont)

- The anchoring effect
  - Avoid off-the-cuff estimates
  - Spend time on your estimates
  - "A date for a date"
- Be willing to overestimate
  - If we are good, should we be over as often as we are under?
- When to commit?
  - Initial high-level architecture (What are the pieces?)
  - The "Last Responsible Moment"
- Manage margin as a resource (to hide or not?)

Estimating Methodology

- Use more than one technique
  - Expert judgment
    - i.e.: analogy, PERT
  - Group estimates
    - i.e.: Planning poker
  - Scenario-based
  - Computing/Decomposition
    - i.e.: features, function points
  - Historical analogy
    - measurements from past projects
    - Actual, not scheduled!!! ☺

Crises: The "Death March"

I love deadlines.
I love the whooshing sound they make as they fly by.

Douglas Adams

What to do?

- No magic bullet
- Create a new schedule with accurate estimates
  - We are looking for reality, not happiness
- Show it to management
  - They won't be happy
- Discuss options
  - Move dates
  - De-scope/defer Features
    - Beware the stair-step effect
  - Add resources*
  - Retire to Hawaii

Summary

- Good software management avoids the "Death March"
- The "Death March" increases technical debt and undermines software quality/reliability
- Work to avoid the "Death March"
  - Identify and manage risk
  - Be mindful of technical debt
  - Improve your software estimates

Additional Reading

- "Software Estimation: Demystifying the Black Art", Steve McConnell
- "Ship it! A Practical Guide to Successful Software Projects", Jared Richardson and William Gwaltney
- "Quality Software Management, Vol. 3: Congruent Action", Gerald M. Weinberg