

Master Solution

This master solution was created by splicing together a number of answers provided by student teams in the class. We selected a subset of answers that were crisp and captured the most essential points. Many other answers were also excellent. Credit is given below.

Project Management Case Studies

Part 1 – Matrix Comparison of Three Cases

Compare all three cases (MS Office 2000, Big Dig, SSN 688i) based on project preparation, planning, monitoring and adaptation.

Credit: Lee & Sermpetis

Page 2

Part 2 – Project Assessment

Give a one paragraph assessment of success and uniqueness of each project. Extract lessons learned for future or similar projects.

Credit: Downen

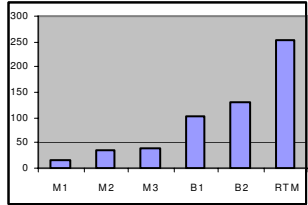
Page 3

Part 3 – Hardware versus Software Development Projects

Summarize your view of the role that hardware and software content play in driving project complexity and effective management techniques. Highlight the particular challenges of projects that have both high hardware and software complexity.

Credit: Text - Buttler & Stevens (Page 4), Table – Barretto & Usan (Page 5)

ESD.36J SPM Homework 5

	Case 1: Microsoft Office 2000 Case (s/w)	Case 2: The Boston Central Artery Project (h/w)	Case 3: 1 st SSN 688i (mixed)														
Project preparation	<ul style="list-style-type: none"> Office 2000 was an incremental improvement over previous versions to address migration, administration (total cost of ownership), html compatibility, web collaboration & reporting, outlook integration and personal productivity Major Stakeholders: Microsoft, Business Users, Individuals Internally funded by Microsoft Project organized in a matrix structure with shared horizontal features within distinct vertical applications. Tripod of program management, coding and test replicated in both horizontal and vertical axis. 	<ul style="list-style-type: none"> Initiated to increase the traffic flow capacity in downtown Boston in a sustainable way. Major stakeholders were the public of Massachusetts, local government, contractors, suppliers, but also the federal state who provided for funding Provide greatest possible traffic relief for future with least possible disruption to the lives and livelihood during construction. 90% funding obtained from highway trust funds Artery tunnel project consists of Cross Harbor Tunnel, Central Artery, and Seaport Access Road. Project is a PPP between the Commonwealth of Massachusetts and Bechtel & Parsons Brinckerhoff. 	<ul style="list-style-type: none"> To upgrade SSN688 to take SSN637 role and incorporate advanced technologies developed Major stakeholders were the US federal government, DoD, contractors. Initial objectives: Enhance SSN 688 class by increasing engine room efficiency, adding under ice capability, adding AN/BSY-1 combat & control, improve sound isolation, and integrated vertical launch system. The PO had 4 main stakeholder parties (DoD, Design yard, Construction yard, and Prime contractors) who control their sub-contractors. 														
Project planning	<ul style="list-style-type: none"> M0: Detailed specs & planning of the next 3 phases, M1,2,3. M3 (code complete) followed by beta1 & beta2. Ready to Manufacture (RTM) scheduled for 7/13/98 (see ex. 4) No external suppliers, internal spiral between coding and testing contributors. Coordination handled by program managers 	<ul style="list-style-type: none"> Major phases: create framework, conceptualization, environmental analysis and construction. Major critical milestones were the north area project that prepared for the larger artery project, construction of the cross harbor tunnel, central artery, and seaport access road, and removal of elevated highways. Coordination responsibilities belong to Commonwealth of Massachusetts and Bechtel & Parsons Brinckerhoff. Physical concept was developed with all stakeholders. 	<ul style="list-style-type: none"> The main goal was to keep GFE off the critical path due to the design agent errors. Main coordination responsibilities lay with the executing PM (SUPSHIP). Methods and tools available were the mainframe and mini-based text schedules, PERT calculations, spreadsheets and word processors. 														
Project execution and monitoring	<ul style="list-style-type: none"> “Check-in” of code, “Private releasing” testing and code reviews within the milestones allowed to ensure quality and keep speed up. Schedule delay in days:  <table border="1"> <caption>Schedule Delay in Days</caption> <thead> <tr> <th>Milestone</th> <th>Schedule Delay (Days)</th> </tr> </thead> <tbody> <tr> <td>M1</td> <td>~20</td> </tr> <tr> <td>M2</td> <td>~40</td> </tr> <tr> <td>M3</td> <td>~50</td> </tr> <tr> <td>B1</td> <td>~100</td> </tr> <tr> <td>B2</td> <td>~130</td> </tr> <tr> <td>RTM</td> <td>~250</td> </tr> </tbody> </table>	Milestone	Schedule Delay (Days)	M1	~20	M2	~40	M3	~50	B1	~100	B2	~130	RTM	~250	<ul style="list-style-type: none"> Maintenance of traffic (pedestrians and vehicles) during construction Safety records Design reviews Poor schedule and cost tracking tools The project met the technical requirements but grossly exceeded its allocated schedule and budget due to increase scope (& cost) and inflation. 	<ul style="list-style-type: none"> Different major stakeholders used scheduling and planning system which was mainframe based with text output and usually structured around either material/cost flow or CPM/PERT. At the integrated implementation level the data are manually integrated and tracked. Daily reviews, design reviews, progress reports, and interacting with workers helped to monitor the project status. It was the norm to have unresolved technical issues, for example initial sea trials were conducted even with 4000 TPRs outstanding.
Milestone	Schedule Delay (Days)																
M1	~20																
M2	~40																
M3	~50																
B1	~100																
B2	~130																
RTM	~250																
Project Adaptation	<ul style="list-style-type: none"> Features to be implemented during each phase were prioritized. Any feature that didn't pass the stage-gates reviews (synchronized across all Office applications) was re-scheduled for the next milestone, replacing lower priority features. This “planned” mechanism of descoping was institutionalized and allowed for relative flexibility and limited delay. 	<ul style="list-style-type: none"> MBTA expanded services to transport more commuters, carpools, expanded harbor services, stricter traffic violations enforcements to keep streets clear. Project increased in scope & resources (eg. High tech traffic control center costing another +\$1B). Management was not effective with the operation and maintenance responsibility. Poor federal and state oversight on environmental, cost monitoring activities. 	<ul style="list-style-type: none"> Upgrading involves integrating new and legacy systems (hardware and software). IBM had to integrate the code packages. Classification of PTRs to assign priority. Initial sea trials after critical issues were resolved Having a rapid prototype “Baby Broadband”. Use magic “Blue Grease” to meet grease line. Re-arranging the layout by waterfront gurus. Complete successfully without descoped and active participation and commitment of stakeholders. 														

PART 2: Post-Project Assessment

Write a short paragraph summarizing your impressions for each project.

What is your summary assessment of this project? Was it an unqualified success? Was it a failure? What are the lessons learned for future projects? Were there follow-up projects? Is the project still on-going? What makes this project unique?

Software Development: MS Office 2000

In most ways this project was a success – a new product with greater functionality (particularly in Internet applications) and integration among the applications (Word, Excel, etc.) was shipped only 8 months late, and Microsoft has since enjoyed steady profits as a result. In a few aspects, however, the project is viewed as less successful, particularly in terms of end-user satisfaction (the product focused on system administrators). Some data also indicates that the software developers were less productive on this project than in the past, perhaps due in part to the rigorous testing process employed throughout the development process (the daily code check-in). One primary lesson to take from this project is that feature descoping can be an effective method for maintaining project schedule in the face of late changes and project challenges. This project was clearly not unique for Microsoft (though many of its management features were) and future software development projects may utilize the lessons learned from the Office 2000 development.

Civil Architecture: Boston Central Artery Project

This project is currently viewed as a qualified success, but in the future will undoubtedly be viewed as a success for the city of Boston, the Commonwealth of Massachusetts, and for the United States as a whole (much as the Hoover Dam and other large, controversial construction projects are now viewed). Many lessons are to be learned from the project (including the effects of political and technical systems interaction), but one of the major lessons is simply that large engineering projects, even with decades of planning and high-level management and political support, will encounter unforeseen challenges resulting in additional financial expenditures and longer schedules than anticipated. This project had a number of unique features, not the least of which is its scope (financial and physical) and the requirements for highly integrated engineering and political cooperation.

Naval Project: SSN 688i

Overall this project appears to have been a success in terms of delivering an intermediate capability to the warfighter before an entirely new submarine class could be developed. Schedule and budget overruns appear to have been minimal with adequate technical workarounds (such as the broadband sonar example discussed in class) developed during testing. Management lessons to take from this project include the necessity to be informed of real progress by being on-site and involved in day-to-day operations as much as possible. An additional lesson is that sophisticated, computer-based management tools are not necessary for successful project management, though they may simplify many aspects or may be useful in identifying critical issues earlier in the program. This project was not unique in that many of the warfighter's tools are upgraded in this manner for enhanced capability (be it a submarine or aircraft, tank, missile, etc.). Lessons and experience from this 688i project should be transferable to other submarine projects whether they are upgrades or all-new designs.

QUESTION #3: Hardware vs. Software Projects

The cases were chosen such that you would be exposed to a spectrum of rather-hardware versus rather-software centric projects. Describe in 200-600 words how the ratio of hardware/software and the complexity of these two aspects drive project management in your opinion. A hypothesis from the Product Development Process lecture was that software projects are amenable to a spiral-development approach, while hardware-development is best done in a phase-gate (waterfall) fashion. Do the three cases support that view?

As discussed in previous homework sets and in other classes, software-only projects have few barriers to rework, and are therefore well suited to a spiral development approach. Software is abstract, and changes to code are usually trivial to accomplish. Since well written software is modular and closely follows function, small changes to that function usually only require changes to one or a handful of modules. The Microsoft team followed a spiral approach, but most Office users would argue that they either got too dizzy or needed to go around a few more times before release. Hardware, on the other hand, is almost entirely form-focused. Changes to that form must not only adhere to the restrictions of space & time, but any preexisting prototypes or parts must be scrapped, or reworked if possible. For these reasons, hardware projects often follow a phase-gate development approach, where portions of the design are frozen at regular intervals to minimize change. The Big Dig followed a phase-gate approach rather well, and it was one of the keys to its success.

It is the hybrid software/hardware projects like the 688i that truly amplify a project's inherent complexity. Since the most suitable management style for each portion of the development is ill-suited to the other, it is a constant struggle to manage the entire project effectively. The physical constraints of hardware usually dominate that struggle. Unless the product is almost entirely software, the hardware is usually developed with a phase-gate approach with the software team spiraling along out of phase behind it, dropping builds at every milestone. Hardware effort in this case is relatively flat until it drops off after critical design review, and software effort starts slow and builds as the hardware design solidifies, reaching a fevered pitch when production hardware becomes available and software/hardware integration occurs.

The danger of this approach lies in compatibility. Unless the hardware is standard or completed early in the design stroke, software is usually not tested on actual production hardware until the end of the project. Several programs at Raytheon discover flaws at this interface only during hardware/software integration testing, and run over budget and beyond schedule spiraling through bug fixes and hardware changes. This is not to say that a phase-gate approach is not the preferable method for a hybrid project – it probably is – but the key to success seems to be managing the interface between hardware and software as if it were an entity unto itself, and making every attempt to begin integration, in any form, as soon as it is feasible.

Table comparison H/W versus S/W project features

Feature	Hardware	Software
Product type	Tangible	Intangible (false sense of project completion)
Development Cycle time	Slow (years or decades)	Fast (months or years)
Lifecycle	Long	Short
Build strategy	Discrete (one step after the other)	Incremental and continuous
Product Evolution	Possible but minimal and occurring slowly	Almost certain and happening rapidly
Prototypes	Difficult to make, time consuming and fairly expensive	Easy, quick and rather inexpensive
Development / testing	Development more important than testing	Development / testing balance

Feature	Hardware	Software
Assets	Capital intensive	Human intensive
Supply chain	Extensive and articulated	Limited and specific
New technology	Can come up and being implement during the course of the project	Privilege off-the-shelf technology. Underestimation of the impacts of new technology can be
Turnover	Manageable	critical
Regulation influence	High	Moderate
Disposal / recycling	A problem to be taken into account into the design process	Not a problem
Public opinion and public authorities involvement	Moderately frequent	Rare
Team members age difference	High	Low