



**ECONOMIC AND SOCIAL
COUNCIL**

Distr.
LIMITED
E/ESCWA/TECH/1999/WG.2/14
6 September 1999
ORIGINAL: ENGLISH

Economic and Social Commission for Western Asia

Expert Group Meeting on Project Planning
and Management in Research and
Development and Quality Assurance
Beirut, 21-23 September 1999

29 SEP 1999
LIBRARY

**TRENDS IN SOFTWARE DEVELOPMENT
(POWERPOINT PRESENTATION)**

Note: This document has been reproduced in the form in which it was received, without formal editing. The opinions expressed are those of the author and do not necessarily reflect the views of ESCWA.

United Nations Economic and Social Commission for Western Asia

Trends in S/W Development - A Project Management Perspective

Frank R. Parth
MicroStrategy, Inc.
University of California - Irvine



Expert Group Meeting on Project Planning and Management
in R & D and Quality Institutions
Sept. 21-23, 1999

1

Trends in SW Development

- General problems with SW project management
- Increasing complexity
- Multiple operating environments
- New development tools
- Changed emphasis to user interface
- Time-to-market
- High development costs
- Requirements management
- New testing needs
- Maintainability
- Alignment to business goals



Slide 2

PM Response to General Problems

- Build processes that make programmers more productive and help them avoid problems.
 - Software Engineering Institute's Capability Maturity Model (CMM)
 - Level 1 - Ad hoc programming/no processes
 - Level 2 - Make the programming process repeatable
 - Do things the same way every time
 - Level 3 - Define the processes
 - Project management, change control, quality assurance
 - Level 4 - Manage the processes
 - Level 5 - Optimize the processes



Slide 5

Trend - Increasing Complexity

- Growth from small single-purpose programs to large, complex programs to programs that run an entire global-wide enterprise.
- Issues:
 - No one person understands the program anymore
 - Increased integration and testing requirements
 - Too many requirements that can be misunderstood or conflict with each other
 - Much harder to manage



Slide 6

Multiple Operating Environments

- Rapid increase in the number of different operating environments
 - Growth in CPU speed and complexity
 - Multiple hardware vendors
 - Compatibility is required with other programs
 - For PCs, this requires compatibility with hundreds of other programs
 - Backward compatibility with earlier versions of the program and with other programs



Slide 9

PM Response to Environments

- Ensure that environment requirements are well defined and documented
- Build a testing and integration program that includes the majority of what the users will be expected to have
 - Make available multiple hardware and software environments
- Include time in the project for this area



Slide 10

Change in Emphasis to User IF

- Old command-line interface is no longer adequate
- Users are no longer computer experts who walk around with pockets filled with notes on how to run a program
 - We don't use slide rules anymore, either
- Users want a Graphical User Interface (GUI)
 - And they want consistency among all the programs they normally use.



Slide 13

PM Response to User IF Emphasis

- The “back-end” processing is no longer the primary part of the program
 - We can't just slap a GUI after we design and code the program
- The User IF is a specialized area that requires a new skill set and design/test approach
 - Make sure you have programmers who can do it
- Design the user IF before (or in parallel with) the back end



Slide 14

Cost Increases

- Costs are escalating rapidly due to:
 - Increased specialization in skills required
 - Increasing size requires more non-programming staff such as integrators and testers
 - Poor documentation means that when a programmer leaves it takes more time for his/her replacement to learn what they were doing
 - Large number of environments requires extensive integration/testing time and hardware



Slide 17

PM Response to Increasing Costs

- The best response to high costs is to lay out the costs as accurately as you can determine them ahead of time, then add 10-20% to allow for the unexpected.
 - If you come back halfway through the project and ask for additional money, you'll be replaced.
- On larger projects, track your costs as closely as you track your time.



Slide 18

New Testing Needs

- New testing needs come from:
 - Increased size and complexity of programs
 - Need to operate in multiple environments
 - User demands for simple, understandable interface



Slide 21

PM Response to Test Needs

- Programmers test their own code at a unit or module level
- An independent testing group tests:
 - Interfaces among modules
 - To multiple operating environments
 - With different users
 - To integrate the program
- Programmers test to meet requirements, testers test to break things



Slide 22

Alignment to Business Goals

- Businesses are increasingly demanding programs that are specialized for their goals.
 - And business goals change rapidly
- Enterprise Resource Planning (ERP) packages
 - Peoplesoft
 - SAP
 - JD Edwards
 - Oracle Financials
 - All require heavy, and expensive, customization



Slide 25

PM Response to Customization

- Requires strong understanding of the business processes and business rules
 - Specialized set of skills that demands additional project members who can understand both the program and the business.
- Extra time and resources required
 - Lots of interviews with the business users
 - Extensive requirements documentation
 - Separate project hardware from normal production system
 - Extensive user testing



Slide 26

Maintenance Requirements

- Programs can no longer be designed in huge packets of millions of lines of code
 - Much too difficult to maintain
 - If you make one change, you have to change everything else to adapt



Slide 23

PM Response to Maintainability

- Modularized code
 - Well-defined interfaces between modules
- Modules can be upgraded without changing other modules
- Separate out the user-interface modules from the back-end modules
- Separate the modules that interact with the operating system from everything else



Slide 24

Requirements Management

- The most critical item in a project is to meet the user's requirements.
 - But you have to know what they are first.
 - Allow 1/3 of the project time to identify, document, and analyze the requirements.
- Most requirements come from:
 - Users
 - Hardware environment
 - Processing environment
 - Other requirements



Slide 19

PM Response to Requirements

- The best approach is to include systems engineers on the team.
- Systems Engineering specializes in:
 - Requirements analysis
 - Performance analysis
 - Integration
 - Requirements change management
- Have somebody do the work, even if you don't call them a systems engineer



Slide 20

Time-to-Market Pressures

- Microsoft has changed the approach of the software industry in getting products to market
 - Get to the market first to gain market share
 - Release a product with basic functionality and only partially tested
 - Use the complaints of the users to determine what to put into Version 2 and where the problems are
 - The first version of Windows that really worked was version 3.1.1



Slide 15

PM Response to Market Pressure

- Early involvement of the project manager to build achievable schedules
- DON'T let the marketing group determine the schedule
 - Demand review of any marketing sent out
- Get approval for the functionality in Version 1
 - Needed functionality
 - Not desired functionality
- Start testing early under multiple environments



Slide 16

New Tools

- SW developers are continually creating new languages and tools
- Ultimately good, however
 - Learning curve
 - Creates new development problems
 - Uncertainty in project plan



Slide 11

PM Response to New Tools

- Try not to use them
- Ensure trained resources are on project
- Add training time for non-skilled resources
 - Programmers, systems engineers, testers
- Use skilled contractors if needed
- Get the tool vendor to help



Slide 12

Example - Increasing Complexity

- In 1984 Apple Computer released MacWord.
 - Program + OS fit on a 700k floppy disk
- Microsoft Word for Windows
 - Begun in 1984
 - Scheduled for completion October 1985
 - Finally released in November, 1989
 - Project size
 - 55 labor years of effort
 - 209,000 Lines of Code
 - Larger than IBM's MVS



Slide 7

PM Response to Complexity

- Managing a large group of programmers is not very different than managing a large group of MEs or EEs.
 - However, because SW is not visually measurable, the best judges of progress are - other programmers
 - Make peer reviews of code an established part of the PM process
 - Include a Systems Engineering group
 - Purpose is to handle requirements and interfaces
 - Change how you measure progress
 - Functional module vs. Lines of Code



Slide 8

General Problems with SW PM

- Progress is not physically measureable
 - You can't count lines of code to see if the task is done
- SW programmers don't like to document
 - And they change companies a lot
- A good programmer is up to ten times more productive than a poor programmer.
 - But you can't tell who is good until the project's end
- Issue: Correct scheduling of critical tasks



Slide 3

PM Response to General Problems

- Project Plan =
- 1/3 = Requirements definition & analysis
- 1/3 = Design and coding
- 1/3 = Test and integration



Slide 4