Effectiveness of Nortel Networks Time to Market (TTM)
ESD.36J System & Project Management

Dec 4, 2003
Team 15
Andrew Bond
Frank Lanni
Mike Nolan
Nicholas Svensson

Courtesy of Andrew S. Bond, Francesco Lanni, Michael Nolan, and Nicholas Svensson. Used with permission.
Overview

• Project Objectives
• Project Approach
• TTM Overview
• MFRM 1 and MFRM 2 Overview
• Traditional CPM
• Design Structure Matrix (DSM)
• System Dynamics Modeling
• Conclusions & Recommendations
It is rare to find two projects, one successful and one unsuccessful as judged by somewhat subjective criteria, performed by the same group of people following the same process. Additionally, the plan of record for each program was readily available as were action registers, schedules, risk decision analysis etc. These two projects provided a fertile test bed for analyses since many of the process and human induced variables were not discriminators between the two projects allowing clearer analysis of endogenous variables.

The business model used by Nortel Networks is significantly different than that of the Defense industry. As the majority of the team members had limited experience in managing a commercial project additional, insight and learning was gained by studying projects whose sole purpose is to bring a product to market quickly without direct customer involvement.
This project was conducted in three phases after gaining project approval and signing nondisclosure agreements with Nortel Networks. Nicholas was the only Nortel Networks employee on the Team and the only member with any technical knowledge of the subject matter. Consequently, the Team’s first phase was dedicated to gaining familiarity, under Nicholas’ tutelage, with the products and their basic function, Nortel Networks’ Time To Market (TTM) process, vernacular and acronyms. The second phase immersed the team in the significant data of similar format for both projects, which required substantial collection and analysis. These tasks were made easier by the superlative data mining efforts of Nicholas Svensson and access to Nortel Networks’ online data repository, Live Link. The team also conducted extensive interviews with the project engineers and program managers of each program to gain further insight. The final phase of the project was to construct the DSM and develop the system dynamics models for the two projects. This final phase significantly overlapped with the second phase. The DSM and system dynamics discussions may be found later in this presentation.
Time To Market (TTM) is Nortel Networks’ product development process (PDP) which was implemented just before MFRM 1 and entrenched by MFRM 2.

Note: TTM has been replaced within the last month by a lifecycle management process due to lack of consistency and product quality across various programs. TTM was intended to instill responsibility and control within the individual project teams, but results have not been favorable. Consequently, Nortel Networks has reverted to a more structured hybrid phase-gate / TTM-like process. The new process also has more of an end-to-end view than TTM by expanding the lifecycle focus.

TTM is not a traditional phase-gate process

• Possible to have groups in different stages of development
• Each group defines their own criteria for what constitutes a business decision point within each project
• The above table describes the activities and deliverables conducted within each phase

Nortel Networks’ training literature describes TTM in the following manner:
“Time To Market is a business model that will enable Wireless internet to:
1. Improve the linkage with customers and the market in general
2. Focus on products/projects/solutions and customers rather than functions
3. Reduce the time it takes to get a product to the market place
4. Improve forecasted revenue growth and business profitability
5. Change the way we make decisions and make decisions that stick”
TTM extensively employs cross-functional teams in developing products and encourages early and continued customer interaction. The wheel structure in lieu of a traditional organization chart is meant to foster more diverse intra-team and customer communication. The spokes of the team structure are the extended team responsible for the functions as described above and may not be exclusive to any one development project. The core team, as shown in blue in the center of the wheel, forms the nucleus of the team and is described in Nortel Networks’ TTM definition as follows:

“A small cross functional team who are given the mandate and responsibility for the introduction and business success of a project. The Core Team works with key customer, and extended team members to ensure that the viability of the project is maintained. The Core Team is responsible to the PMT (Portfolio Management Team) for the overall business success of the project from inception (prior to Market Readiness) until its disbanding (at or after Channel Readiness)

Core Team responsibilities include:

• Establishes a contract and charter with the PMT which describes financial, performance and project schedule, product characteristics and cost
• Works with PET (Product Evolution Team) to ensure consistent with roadmap
• Manages end to end introduction of the product or solution
• Work with lead customers to ensure that the program/solution remains viable (A change in the viability of the program results in an Out of Bounds condition)
• Collaborates with functional manager for successful program execution
• Communicates regularly to the organization on progress and issues
• Manages activities/results of the extended team through the Core Team members
• Initiates risk analysis and mitigation activities
• Prepares BDP (Business Decision Point) presentations to PMT”
Common Factors
As research progressed, the team confirmed the similarities between the two projects as listed above. Not all of the items listed were identical, but the differences were either negligible or the change was easily modeled or understandable.

Differences
The team also discovered that there were significant exogenous variables that significantly impacted the projects. Addressing these variables proved to be very difficult. The two major exogenous factors to be addressed were dynamics within Nortel Networks as a company and the broader cellular phone market.

Health of Nortel Networks as a company
During MFRM 1’s development (the less successful project), Nortel as a company was riding high, which induced some level of distraction. Additionally, hiring was rampant and salaried employees were paid overtime. Conversely, Nortel was suffering corporately during the MFRM 2 project (the more successful of the two). There were significant reductions in force and salaried employees were no longer paid overtime, but morale within the project remained high.

Market
MFRM 1 was launched during a boom in the cellular phone industry and Nortel Networks’ customers wanted the product as soon as possible and were willing to pay for a product of inferior quality. Market pull made schedule the driving criterion for customer acceptance. Additionally, no one really understood the ultimate size or equilibrium point of the cellular phone market.

MFRM 2 was a cost reduction program and was launched during a slump in the cellular phone industry when Nortel Networks’ customers had begun to understand the size of the market and would only accept the product if the quality was high and costs were lower. Consequently, MFRM 2 was pushed on the marketplace, so cost was the most important criterion.
The two Multiple-Carrier Flexible Radio Module (MFRM) projects, as previously discussed, are essentially base station radios for carrying commercial cellular phone traffic and are produced in several frequencies.

The project priority matrices, as introduced by Kos Ishii of Stanford University’s Manufacturing Modeling Laboratory in the System Architecture class, demonstrate the fundamental differences between the two projects with regard to technical scope, cost and schedule. It can be seen that the driving constraint on MFRM 1 was schedule followed by technical features and Nortel Networks was willing to accept the standard (recurring) cost. Conversely, on MFRM 2 standard (recurring) cost was the driving constraint followed by technical features and finally schedule.
This slide summarizes some of the differences and similarities previously discussed and introduces several new points. Several of the major differences are shaded red and are fairly self-explanatory. However, the un-shaded “multi-site program” is a significant factor and is discussed in more detail on the MFRM Value Streams slide. It should be noted that both projects had durations of approximately 24 months. MFRM 1 was typically staffed with about 65 people involved in hardware design and 12-15 software developers. MFRM 2 had a lower typical staff of approximately 50 hardware people and the same 12-15 software developers.
Lots of Concurrent tasks
Tough to visualize scope of risks taken
Identifies critical path but risk in concurrent tasks could quickly change critical path
Harder to identify coupling than in DSM
Taken from the MS Project plan shown in previous chart.
204 coupled tasks.
Large loops and dependencies exist particularly between Beta and customer final acceptance testing and completion of Alpha and product definition.
Partitioning showed logical errors in the program schedule used by the core team. Large coupled block at level 3 represents the Alpha and Beta development cycles, level 5 was started before level 4 was completed, the short duration of the Strategic Readiness phase is also shown indicating the limited planning.

In MFRM1 the undiscovered rework from Beta became apparent after channel ready and therefore a very large non accounted for iterative loop occurred at the end of the project.

On MFRM2 the Beta phase was delayed due to the more thorough verification of the product to the customer making channel ready occur without incident.

MFRM2 is viewed as a higher quality product than MFRM1. It is also viewed as costing more although this is not entirely true as life cycle costs are measured differently than R&D costs within Nortel.
DSM Conclusions

- Tasks at lower levels were begun before completion of the previous level (a.k.a. coupled block) e.g. Alpha started before Business Readiness achieved. Beta phase started before Alpha complete. High risk for undiscovered rework.
- Taking risks introduced additional unexpected rework cycles and collateral damage amongst constituent tasks in multiple levels which were not taken into account (assumed the best-case scenario)
- Although the Core Team was cross-functional the rest of the organization was not. Fewer artwork spins may have been avoided through co-location of the multi-site teams during these phases.
- Tasks inside each coupled block were not effectively worked in a concurrent manner for MFRM1 much better on MFRM2.
- Risk, task duration and good judgment must be applied to DSM results to optimally plan the project
5 Major Factors evoking system dynamics for MFRM1 & MFRM2

- **Nortel Stock Price Gradient**
  - MFRM1: times are good, people are distracted by portfolio values, quality and productivity suffer, Nortel couldn’t hire people fast enough
  - MFRM2: telecom bust, massive layoffs, quality becomes the key differentiator between Nortel and competitors

- **Technology**
  - MFRM1: learning curve yet to climb, lack of experience in critical areas
  - MFRM2: seasoned staff from MFRM1

- **Staffing Strategy & Risk Stance**
  - MFRM1: Conscious decisions to ignore, or postpone risk mitigation activities to later phases of the program. Transition to different group with smaller staff prior to discovery of rework
  - MFRM2: No voluntary postponement of risk mitigation activities. Delay transition until rework is discovered and dealt with.

- **Concurrent Task Execution**
  - MFRM1: Conscious decision to start iterative tasks in parallel – multiple baseline chaos
  - MFRM2: Conscious decision to start iterative tasks in parallel – controlled baselines

- **Internal Push / External Pull**
  - MFRM1: The customer was screaming for the product so they could address their own competition. In an effort to be responsive, put the “quick-and-dirty” solution out there for the customer, warts and all. Time to market was more important than quality.
  - MFRM2: The market would not bear poor quality, but estimated each week of schedule cost between ½ and 1 million dollars

Well known or common effects modeled in the ESD.36J Rework model were in play for some of the Nortel project dynamics in MFRM1 & 2

- Experience on quality
- Schedule pressure on quality and productivity
- Willingness to hire, and time to gain experience
- Fraction perceived complete and staffing
- Prior quality on quality – but this is an inadequate proxy for effect of concurrent task execution – seek to modify
System Dynamics Model vs. Actual

Actual Work Done

Simulated


0
System Dynamics Model vs. Actual

Peace Shield Results

Actual Work Done

Simulated

Let’s take a look at the model components

- We start with Work Done
  - Nortel used an iterative development process, with alpha, beta and clean up releases
  - We felt this was a valid process and that work done in the early iterations represented work done and not rework
    
    Even with perfect quality, iterations should occur
We started with the same major structural blocks as the model we used in class:

- Work To Do
- Undiscovered Rework
- Work Done
We started with the same major structural blocks as the model we used in class:

- **Work To Do**
- **Undiscovered Rework**
- **Work Done**

The model used in class assumed the initial spins of a multi-spin project were rework. We modeled them as Work Done. Spins are a necessary part of technological product development. Even with perfect quality, spins should occur.

DSMs show iterations occur and problems in one area result in having to redo work in another area. This type of rework was also evident in the projects at Nortel.

The Nortel teams made decisions on how much development to do concurrently and how much to do serially. Learning from Nortel is important since concurrent development behaves differently than serial development.

Concurrent development results in significantly more risk of collateral damage than serial development.

We define collateral damage, or knock-on effect, as a work task done according to plan that doesn’t contribute to progress because of a problem with another task.

An example would be a circuit board built according to plan using a defective circuit design. The building of the board was correct, the quality good, but no benefit gained due to the faulty design.

Our study provides an excellent first look at the relationship between DSM and systems dynamics and should provide project management teams a better feel for the dynamics involved between the degree of concurrent development, quality, and iteration cycle size.

In MFRM 1, significant concurrent development was accepted when they made the decision to send hardware out to the 9 locations without having a baseline. So they were concurrently fixing the hardware at the same time they were developing MTX10 software.

Towards the end of MFRM 2, significant concurrent development risk was accepted due to the time to market pressure.

In the end, the level of concurrent development probably slowed them down.

Need to be able to back this up with specifics.

So, starting from the left side of the diagram, the fraction of work done concurrently effects the staff level.

For example, if they originally planned to work 40 people for 2 years and then changed the plan to do 100% of the work concurrently, effectively the staffing would change to 80 people. If all the tasks were independent, they could complete the job using 80 people in 1 year.

Work then flows from work to do into either tasks started concurrently or tasks started serially.

- From a task started, one of three things can happen:
  - It can be completed, flowing to Work Done.
  - It can flow to Undiscovered Rework if there is a quality problem and it is not done.
  - Or it can become undiscovered rework due to collateral damage.

This is true both of tasks started concurrently and tasks started serially.

[i] Or stabilizing

-------

Majority of rework was impact on tasks done correctly by tasks done incorrectly.

Very small fraction of the tasks were purely defective.

Off-nominal tasks aren’t always due to just one cause.

Partially technological difficulty factor, partially due to schedule pressure, and partially due to other factors.

Judgment whether a task is considered 100% done, or partially done and partially undiscovered rework.

MFRM1 took five iterations to get the product ready for market.

1st iteration was an incorrect technological path due to technology risk, inadequate planning and risk assumption.

Learning from 1st iteration added some value so not a total loss.

Serial Tasks

Concurrent Tasks
DSMs show iterations occur and problems in one area result in having to redo work in another area. This type of rework was also evident in the projects at Nortel. The Nortel teams made decisions on how much development to do concurrently and how much to do serially. That is important to model since concurrent development behaves differently than serial development. Concurrent development results in significantly more risk of collateral damage than serial development.
We define collateral damage, or knock-on effect, as a work task done according to plan that doesn’t contribute to progress because of a problem with another task.

An example would be a circuit board built according to plan using a defective circuit design.

The building of the board was correct, the quality good, but no benefit gained due to the faulty design.

From a task started, one of three things can happen

- It can be completed, flowing to Work Done
- It can flow to Undiscovered Rework if there is a quality problem and it is not done
- Or it can become undiscovered rework due to collateral damage.

This is true both of tasks started concurrently and tasks started serially.

Our study provides an excellent first look at the relationship between DSM and systems dynamics and should provide project management teams a better feel for the dynamics involved between the degree of concurrent development, quality, and iteration cycle size.

In MFRM 1, significant concurrent development was accepted when they made the decision to send hardware out to the 9 locations without having a baseline. So they were concurrently fixing the hardware at the same time they were developing MTX10 software.

Towards the end of MFRM 2, significant concurrent development risk was accepted due to the time to market pressure.

In the end, the level of concurrent development probably slowed them down.
So, starting from the left side of the diagram, the fraction of work done concurrently effects the staff level.

For example, if they originally planned to work 40 people for 2 years and then changed the plan to do 100% of the work concurrently, effectively the staffing would change to 80 people. If all the tasks were independent, they could complete the job using 80 people in 1 year.

Work then flows from work to do into either tasks started concurrently or tasks started serially.

Extra:

Majority of rework was impact on tasks done correctly by tasks done incorrectly
Very small fraction of the tasks were purely defective
Off-nominal tasks aren’t always due to just one cause
        Partially technological difficulty factor, partially due to schedule pressure, and partially due to other factors
        Judgment whether a task is considered 100% done, or partially done and partially undiscovered rework.
        MFRM1 took five iterations to get the product ready for market
            1st iteration a was an incorrect technological path due to technology risk, inadequate planning and risk assumption
            Learning from 1st iteration added some value so not a total loss

Serial Tasks
Concurrent Tasks
This shows that as the fraction of work done concurrently increased, the time to complete decreased, then increased.

The Total project cost was slightly lower at a fraction of .05 than .02, but as the fraction of work done concurrently increased, cost increased since more rework had to be accomplished.

Again, the data put into the model was our best estimate based on a top down view of the program since detailed data were not available. This system dynamics model provides insight into the relationship between the factors but is not a predictive tool.
Information was gathered in the form of data from Nortel project databases and interview responses. Quantitative information included beyond the MSProject (CPM) Gantt charts included budgeted vs. actual costs (by month) for MFRM1/2 for 24 months and extensive lessons learned log for MFRM1/2.

The MFRM1 lessons learned data provided good source of questions for interview purposes, and served to reinforce, corroborate, or raise questions about responses given in interview sessions with Nortel project team members.

Qualitative information was extensive, but quite difficult to transform into quantifiable modifications, or calibrations of the ESD.36J re-work model parameters.

- Success for the projects was primarily based on getting to market quickly, which is somewhat intangible, and also considered factors of quality and life-cycle cost.
- Planned iterations are an inherent part of the development process at Nortel, and as such determination of rework vs. work done becomes a somewhat subjective matter.
- Morale as impacted by the financial state of the company is also very difficult to quantify.

The prior work quality on quality effect did not adequately represent work completed correctly, but rendered invalid, as a result of missing/assumed information from influencing tasks that are executed concurrently. As such the “collateral damage” modifications have been proposed. Model tends appear to be correct, but complete calibration is beyond the scope/schedule of the current effort, and is left as an exercise for Nicholas and his coworkers.

Other effects modeled in the ESD.33J Rework model that were addressed in the modeling of MFRM1 & 2 were:
- Experience on quality
- Schedule pressure on quality and productivity
- Willingness to hire, and time to gain experience
- Fraction perceived complete and staffing
Conclusions & Recommendations

- MFRM1 characterized by high risk, poor performance, and low nonrecurring cost
- MFRM2 characterized by low risk, good performance, higher nonrecurring cost
- Significant differences between MFRM1 and MFRM2 still result in similar schedules
  - Differences were drivers on quality and cost
  - Similarities were the drivers on schedule
- Iterative/Parallel tasking is the critical factor for schedule
  - Significant number of interrelated tasks shown in DSM
  - Explicit modeling of Collateral task execution warranted

"No matter what we do it always seems to take 2 years to complete"

Nicholas Svensson, Senior Manager Supply Chain Operations

Notes:
Cost refers to development costs
Performance refers to quality
Risk refers to program technical risk

Program differences such as experience level, company financial health, technology, risk stance and push/pull factors were drivers on cost and quality.
Program similarities such as the level of concurrent or parallel work, were presumably the drivers on schedule. DSM helps to quantify the number and extent of parallel task execution in the two programs.

For summary figures:

Qualitative
Green = good
Yellow = ok
Red = bad

Pseudo-Quantitative
Small
Moderate
Large
Conclusions & Recommendations (Cont’d)

• DSM may help influence risk assumption in the future
  – Iteration potential and impact
  – Adding a gated process would reduce the groups within each block and potential for iteration

• System Dynamics model was useful for predicting trends and confirming intuition but the process was too complex to produce quantitatively accurate yet reasonably simple model

• Concurrent tasks and iterations indicated by the DSM proved problematic to model in system dynamics

• Use of DSM and system dynamics provided a more integrated view of a complex development process
This slide illustrates the major value streams or interaction streams for MFRM 1 & 2, which can be grouped into other projects, CMs (Contractor Manufacturers – companies that build Nortel designs for them) / OEMs (Original Equipment Manufacturers) and Nortel itself. The arrows indicate interaction and iteration among the bodies and the thickness of the arrows illustrates the relative intensity of the interaction/iteration. (e.g., MFRM 1 experienced a lot of iteration with the CMs/OEMs whereas MFRM 2 experienced iteration but to a lesser extent.) It can be seen that MFRM 1 had more “moving parts” and more interaction/iteration. MFRM had fewer moving parts, which helped increase the “clockspeed” and make the project more efficient and successful.

**MFRM 1**

*Other Projects*
- MFRM 1 was dependent on successful deployment of MTX10 software, software that controls the cellular phone network, before MFRM1 could be fully tested and deployed.

*CMs/OEMs*
- The circuit boards were directed to be subcontracted to Solectron by Nortel Networks management. This proved to be a problematic relationship as Solectron was formally owned by Nortel Networks and it created good, bad and uncomfortable team dynamics. Also Solectron’s boards had significant quality problems.
- Subcontracting the power amplifier added complexity to the project by adding another set of variables, interfaces and bureaucracies to the project.

*Nortel*
- The hardware and software development was not collocated as some of each development was done in both Calgary and Ottawa, which caused significant communications, logistics and technical issues.

**MFRM 2**

*Other Projects*
- MFRM 2 was not dependent on any other projects.

*CMs/OEMs*
- The circuit boards were subcontracted to Celestica on a best value criterion through a competitive process. Relations with Celestica employees were more professional than they were with Solectron and the quality of the product was much higher.

*Nortel*
- The MFRM 2 power amplifier was designed by Nortel Networks and, even though it was built by a contractor, it was far more successful than the MFRM 1 power amplifier.
- The hardware and software development was not collocated as some of each was done in both Calgary and Ottawa, which caused communications, logistics and technical issues. However, the roles and responsibilities on the MFRM 2 project were better defined, which mitigated many of the problems experienced on MFRM 1.
Here we’re trying to convey the feeling of confusion and complexity involved in collecting systems dynamics data for this project, so if this slide gives you a headache, it’s served its purpose.

-We’re focusing on the work to do shown in the red oval

  -So we thought we were asking a simple survey question

    -Did the work get done or did it end up as rework

    …and you would not believe the mealy mouthed excuses these Nortel people could come up with for not being able to answer this simple question

-So we categorized most of their excuses and although I’m sure you can’t read the print, you can get the idea that there’s a lot of complexity

  -There’s the concurrent/serial issue

  -There’s the it seemed to be useful for the alpha release issue

  -There’s the issue of partial failure

  -There’s the issue of whether or not it could have been foreseen (or whether it was just one of those dead end roads you end up having to explore in order to find the right way)

  -Ect.