Time to Market - Solving the Problem of Viable Processes Through Research

by Assoc. Prof. Karl Reed, FACS, FIE-Aust., MSc, ARMIT
Director, Computer Sys. & Software Engineering Board, ACS,
Member of the Board, SEA-Vic.,
Member, IEEE-CS/ACM Software Engineering Coordinating Committee,
Department of Computer Science and Computer Engineering, La Trobe University
Hon. Visiting Professor, Middlesex University

“time to market is shorter than time to design…
and time to evaluate ROI” reed
This Presentation will cover..

1. The Problem and Its Business Implications,

2. Standard Software Processes, Their Limitations and Advantages

3. Time-to-Market (TTM) as a truncated schedule..the impact

4. Business Characteristics needed from TTM

5. Current State of Knowledge and Practice-MS vs the Rest

6. A Research Program for Practical TTM Processes

7. Conclusions
Objectives of this Presentation..

- Analyse the Problem,
- Demonstrate that Researchers can Assist the Software Industry
- Propose a credible work-plan,
- Form a working Party...
1. TTM- The Problem and Its Business Implications

“TTM-the ability to project software into a market in abnormally short time-scales..”

The Problem…

- How to do this and make a profit..
- How to apply to non-shrink-wrap, multi-sale environments
1. TTM- The Problem and Its Business Implications (cont’d..)

- The Business Implications.. TTM allows..
  - Market pre-emption… product to market ahead of competitors (cf Microsoft and Netscape),
  - Rapid response to evolving customer need, in rapidly evolving domains
  - Customer “lock-in” through pre-mature release..
  - Virgin markets to be “captured”
  - Fixed, immediate deadlines to be met..
1. TTM-The Problem and Its Business Implications (cont’d..)

⇒ TTM is not new.. Or unique to s/w..

⇒ The IBM OS/360 … (~1963-66)..  

(... productivity may be 7 times better for a small team, and personal productivity may be 7 times better, so a 5000 person year project might be completed in 10 years by 10 people.. “.. But would anyone be interested…” ) Brooks 1975

⇒ The auto industry operates in a TTM environment
2. Standard Processes - their limitations and advantages

- Limitations…
  - relatively slow
  - delay functional demonstrations
  - schedule not an independent variable
Advantages

- schedule and effort determined by task in hand and resources
- quality outcomes controlled
- functional coherence on complex projects possible(?)
- excellent for “custom builds” and “in-house” projects.. “oncers”
- productivity “high”
3. Time to Market as a “truncated schedule”, uses and impact

- Delivery schedules severely truncated ..(< than 50% of normal) Schedule fixed by external factors..

- Used where there is..
  - Ultra-high volumes of sales
  - Need for market capture (see earlier)

- Impact
  - Productivity reduced (?)
  - Quality reduced(?)
  - Functional Coherence reduced
  - Slow, unreliable, incomplete
3. Time to Market as a “truncated schedule” …… (cont’d)

- The productivity problem
  - for a given project
    - a minimum feasible delivery time $t_{dmin}$
    - cost = $k t^{-4}$ (Putnam’s SLIM)… cost increases dramatically with reduced schedule
    - cost increases with project staff-size
3. Time to Market as a “truncated schedule” …… (cont’d)

The Quality Problem

-Thorough design of high-functionality systems takes time, especially if there are complex HCI interactions,

-Complete testing of high-functionality systems takes time

-Error-free code takes time…

-Efficient code takes time…

-(re-useable code/design takes time..)
4. Business Characteristics of TTM Projects...

- Hard delivery dates
  - year 2000, GST cut-over
- Finite windows of opportunity
  - Web-development technology
- Pre-emptive products
  - beating Microsoft to the market
- Leveraging new technology
  - net-centric products (*net assembled and delivered*)
4. Business Characteristics of TTM Projects...(cont’d)

- Projects where initial sales provide future market dominance
- Ultra high volume products
  - sale price almost independent of development cost
- Time-critical request for proposals providing a bidder with a competitive edge
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Standard Development (under control)</th>
<th>Time to Market (current situation)</th>
<th>Time to Market (target situation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>controlled</td>
<td>truncated</td>
<td>truncated</td>
</tr>
<tr>
<td></td>
<td>acceptable</td>
<td>unpredictable</td>
<td>predictable</td>
</tr>
<tr>
<td>Cost</td>
<td>controlled</td>
<td>unacceptable</td>
<td>acceptable</td>
</tr>
<tr>
<td></td>
<td>acceptable</td>
<td>high</td>
<td>predictable</td>
</tr>
<tr>
<td>Quality</td>
<td>high</td>
<td>usually low</td>
<td>high</td>
</tr>
<tr>
<td>Customer satisfaction</td>
<td>high</td>
<td>usually resentful</td>
<td>high</td>
</tr>
<tr>
<td>Runtime resources</td>
<td>minimal</td>
<td>excessive</td>
<td>predictable</td>
</tr>
<tr>
<td>Design quality</td>
<td>high</td>
<td>poor</td>
<td>high</td>
</tr>
</tbody>
</table>

Karl Reed sea 2000
Time To Market
5. Current State of Knowledge and Practice-MS vs the Rest..
Clearly there are “processes” that work..

- Microsoft has made the TTM concept famous,
  - seems a little like the old “Mongolian Hords”, adhoc approach
- Major telco’s claim success based on COTS approaches.. MCI claimed it was shooting for about 10 weeks for major product deployment
Dijkstra and THE OS in the 70’s (the lesson?)…

“Five people as smart as Edgar Dijkstra can do anything” Reed, 1981

The first Unix effort…(but what did it take for product versions)

OS/360, PL/I (the lesson?) (60’s)...

Very large teams can build large systems very quickly..~ x1000 person years

Total volumes of functionality (e.g. OS/360) may allow partitioning..TTM issue obscured..
5. Current State of Knowledge of TTM...

- The $64,000?…How does MS do it?
  - “synchronise and stabilise”

- Netscape?
  - modified “synchronise and stabilise”

- OSA?
  - Pretty much the same…

- Issue.. What is really new?
Microsoft’s “Synchronise & Stabilise”

[Cusumano & Yoffle 1999] fig 1

Karl Reed sea 2000
Time To Market

ieee-cs computer v38 n10 Oct. 1999
**Microsoft’s “Synchronise & Stabilise”**
*(cf waterfall)*

<table>
<thead>
<tr>
<th>Synchronize and stabilize</th>
<th>Waterfall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification, development, testing are done in parallel</td>
<td>Phases are completed sequentially</td>
</tr>
<tr>
<td>Vision statement is created and specification evolves (spec is output, not input)</td>
<td>“Complete” specification document and detailed design is done before coding</td>
</tr>
<tr>
<td>Prioritized features are built in three or four milestones</td>
<td>All product pieces are built simultaneously</td>
</tr>
<tr>
<td>Synchs are done frequently (daily builds), with intermediate stabilizations (milestones)</td>
<td>One late and large integration and test phase occurs at the project’s end</td>
</tr>
<tr>
<td>Ship dates are “fixed,” but there are multiple release cycles</td>
<td>An attempt is made to achieve feature and product perfection</td>
</tr>
<tr>
<td>Customer feedback is considered during development</td>
<td>Customer feedback serves as input for future projects</td>
</tr>
<tr>
<td>Large teams work like small teams regardless of project size</td>
<td>Many individuals work in large functional groups to scale up projects</td>
</tr>
</tbody>
</table>

## Netscape Staff Allocations for TTM Projects

<table>
<thead>
<tr>
<th></th>
<th>Client products</th>
<th>Server products</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software engineering</td>
<td>110</td>
<td>200</td>
<td>310</td>
</tr>
<tr>
<td>Testing (QA)</td>
<td>50</td>
<td>80</td>
<td>130</td>
</tr>
<tr>
<td>Product management</td>
<td>50</td>
<td>42</td>
<td>92</td>
</tr>
<tr>
<td>Subtotal</td>
<td>210</td>
<td>322</td>
<td>533</td>
</tr>
<tr>
<td>Other*</td>
<td>30</td>
<td>98</td>
<td>128</td>
</tr>
<tr>
<td>Total</td>
<td>240</td>
<td>420</td>
<td>660</td>
</tr>
</tbody>
</table>

*Persons in activities such as documentation, user interface design, OEM porting, internationalization and localization, and special product support.
5. Microsoft’s/Netscape “Synchronise & Stabilise” .. Non-standard features..

- “Daily” build and testing,
- High-leverage tool usage NOT reported
  - Development technology not useful or irrelevant
- Feature instability.. 
  - Features responsibility of teams, which may change them..(?)
  - Mid project major enhancement (NS, MS)
- Automated reporting and testing
  - Build-cycle times ~ a few hours
5. Microsoft’s/Netscape “Synchronise & Stabilise” .. Non-standard features..(cont’d)

- Pre-emptive QA seems “light” (not clean-room)
  - Code reviews not intensive or mandated
  - NS-change control and altered-code reviews, sign-off and recording

- High reliance on developer motivation and progressive testing
  - feature-complete testing precedes β .. fix acceptance controlled during β

- Macro-process independent.. Individuals do whatever
We identify five issues...The technology:

a. embodied by the complete product,
   Its capabilities, in function, performance etc.

b. of its components,
   the parts, materials, sub-systems

c. used in its production,
   tools, design techniques

d. in terms of human skills
   used in production,

e. used in the process of production
   organisation of production, process
5. Microsoft’s/Netscape “Synchronise & Stabilise” .. Non-standard features .. (cont’d)

- Bug tracing during stabilization extensive ...

*NOT REPORTED BY OBSERVERS* ..

- formal documentation requirements …
- conventional phase-completion checks
- feature conformance requirements for OS api’s
- feature coherence testing and checking
- Compatibility control ..
6. A Research Program for Practical TTM Processes

Our goal...

Traditional Process Attributes…
Quality, Cost, Performance, Functionality, Predictability

+ Truncated Schedule

Australian T"ime To Market Methodology
Research Questions

Schedule
- What is an optimal/reasonable project schedule?
- Is it feasible to attempt a project within a specific timeframe?

Resources
- How can available human resources best be allocated to ensure projects succeed?
Research Questions

- Factors determining schedules
  - estimating
  - staff quality and experience
  - methodology
  - re-use
  - tools
Research Questions
Determining candidate markets

Product opportunities what else is a candidate?

- identification of opportunities and time frames assessed
- what project aspects should be emphasised to gain maximum marketing leverage?
- technology and market prediction
- what technology will be market ready when needed?
Research Outcomes.. What do we already know?

Prototyping has several elements....

- eliminating non-coding activities
- relaxing performance constraints
- simplifying error recovery
- controlling UI complexity
- using highly leveraged development domains
- emphasis on interfaces and skeletons

Incremental enhancement of the above
Impact of experience and team cohesion

- **Continuity of Experience Syndrome (Reed 76)**
  - Repeated experience gives massive gains in productivity

- **Team consistency and coherence**
  - Experienced teams share corporate knowledge

- **Domain and Platform experience**
  - IBM study showed these more important than technology
  - Specialists in narrow fields are very highly productive
  - Moving to a new domain/platform could incur learning curve losses
Research Outcomes.. What do we already know?

- Impact of experience and team cohesion
  - The above leads to formal and informal “patterns”

- Methodology Myths…
  - Detailed methodologies are best..
    - But we know they are not followed..
  - Rigid frame-works will be best..
    - See above, what is the cognitive model valid?
  - Documentation aids design…
    - If so, why is it hard to have it happen?
Research Outcomes.. What do we already know?

- Methodology Myths...(cont’d)
  - Specialists at each phase of (Waterfall) process is best, with work product handover..
    - But we know (~60’s) that allocating vertical slices of design, coding, testing yields better code faster..

Part of the Problem.. What do we really know about successful s/w development?

Not enough!
"Extreme programming"?

No need for “third-party” readable work products!

“Extreme programming”?

Private s/w process? (PeSP compliant?)

Optimal task allocation, observed <1970 one or two people
**Desired Research Outcomes**

- **Tools**
  - development tools

- **Methodologies**
  - time to market focus

- **Management strategies**
6. A Research Program for a Practical ausrotam

- Determine existing best practices
  - Collaborative work with institute partners from industry and academia
- Develop models, methods, tools and methodologies
Research Team Activities

- Assess models, methods and tools
  - field assessments using real projects
  - internal assessments using institute projects
- Disseminate knowledge gained
  - field application of methods and tools with institute partners
  - publication
<table>
<thead>
<tr>
<th>Issue</th>
<th>Factors</th>
<th>Approach</th>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Impact of schedule reduction</td>
<td>Data collection</td>
<td>Calibrated new estimating techniques</td>
</tr>
<tr>
<td></td>
<td>Risk assessment</td>
<td>Process recording exemplar projects</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Codification of known models (eg Microsoft)</td>
<td></td>
</tr>
<tr>
<td>Resource allocation</td>
<td>Impact on project planning</td>
<td>Tool assessment</td>
<td>Identification of risk indicators</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data collection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Process recording</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Impact of decomposition models and parallel implementation</td>
<td></td>
</tr>
<tr>
<td>Staff</td>
<td>Quality</td>
<td>Skill identification, fine grained classification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Experience Identification and recording of experience</td>
<td>Improved selection techniques</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project structure</td>
<td>Training need identification</td>
<td></td>
</tr>
</tbody>
</table>
## Research Strategy

<table>
<thead>
<tr>
<th>Issue</th>
<th>Factors</th>
<th>Approach</th>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Improved productivity</td>
<td>Analysis RAD/JAD</td>
<td>High quality prototyping</td>
</tr>
<tr>
<td>Re-use</td>
<td>Areas of application</td>
<td>How to achieve this?</td>
<td><strong>Components</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>What is re-use?</td>
<td>Plans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assess current re-user practices</td>
<td>Designs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Importance of experience</td>
<td>Test plans</td>
</tr>
<tr>
<td>Tools</td>
<td>CASE versus lightweight</td>
<td>What do we really need here?</td>
<td>Tool set selection</td>
</tr>
<tr>
<td></td>
<td>Project management and planning</td>
<td>Tool integration</td>
<td>New tools</td>
</tr>
<tr>
<td></td>
<td>Languages</td>
<td>Analysis of existing tools</td>
<td>Choice</td>
</tr>
<tr>
<td></td>
<td>Process/design recording</td>
<td>Review and recommendation</td>
<td>Identify processes</td>
</tr>
</tbody>
</table>
**Time to Market Conclusion**

- Massive competitive advantage from time to market products with traditional high quality and predictability

- Necessary elements of time to market focussed processes may be accessible—we may already know the answers

- Detailed ‘research’ by experienced staff with access to current practice is most likely to be successful
Time to Market Conclusion

Thank You…

Acknowledge input from..

david cleary, paul radford (chrismatek), dan marantz, jason baragry

Sea working group on TTM

⇒ A working party to be formed...