Collaborative Software Design & Development

*Coordination*

Dewayne E Perry
ENS 623A
Office Hours: T/Th 10:00-11:00
perry @ ece.utexas.edu
www.ece.utexas.edu/~perry/education/382V-s08/
Coordinating Teams

- Multiple locations for teams
  - Separate floors, buildings
  - Separate locations (e.g., Oak Hill and Parmer Lane)
  - Separate states, countries, time-zones, cultures

- Increasing globalization
  - Labor costs
  - Round the clock development
  - Business needs
    - Required local offices
    - Required native employees

- Geographical and temporal separation
  - Informal, unplanned, ad hoc communications critical in supporting collaboration and teamwork
  - As size and complexity of the software system increases, the need for informal communication increases dramatically
Coordinating Teams

- Tom Allen: Managing the Flow of Technology
  - Conflicting goals that need to be met
    - The activities of the various disciplines and specialties must be coordinated to accomplish the goals of multi-disciplinary projects
    - Projects must be provided with state-of-the-art information in the technologies they rely on
      - Best accomplished though face to face communication
  - Functional organization
    - Historically oldest
    - Disciplines/specialties are grouped together
    - Appropriate where fast pace of change
  - Project organization
    - Single point of contact
    - Appropriate where duration of project is long

- Bases for coordinating geographically projects (Lucent)
  - Functional areas of expertise
  - Product structure
  - Process steps
  - Customization
Coordinating Teams

Functional areas of expertise

What: expertise localized

Benefits:
- Better load balancing
  - Smaller set, availability
- Better way to develop/enhance expertise
  - Mentoring, keeping up to date

Problems:
- Critical: assignment of responsibility
  - Feature management
- Unexpected events tend to be disruptive
  - Hard to find appropriate support

Coordination needs
- Co-location
- Accurate project plans
- Detailed development process
Coordinating Teams

Product structure

What: organize along architectural structure
- Optimal: one component at a single site
  - Need good interfaces

Benefits: independent operating environments
- Don’t need same processes, environments
- Do need standards, defined interfaces
- Don’t need extensive retraining of acquired companies/people
- Accommodates different countries, cultures

Problems
- Local unit testing works well – local expertise
- BUT system manager – spans all components – no local expertise – experts dispersed
- One feature, many components
  - Originally separate components
  - Now a unified product with coordination across components

Coordination:
- Co-location helps
- Standards and interfaces
- Defined milestones to coordinate development
  - Eg, originally 3 different project management systems
  - No milestones in common between the three
Coordinating Teams

Process Steps

What: separate steps - handoffs between steps

Benefits:
- Closer to the customer
  - Useful if market is for multiple markets
  - Deployment critical for customer satisfaction
- Better use of scarce resources
  - Eg, expensive test labs

Problems:
- Temporal dependencies - handoff delays, different priorities
- Handoff points
  - Need clear agreement - eg, 100% tests
  - Person along with release across geographical boundaries

Coordination:
- Clear, agreed to specifications for trade-offs
- Stable plans for dates
- Shared understanding of development processes
Coordinating Teams

 withStyles 1

 Customization

 What:
 - One site has the core code
 - Other sites do customization for market specific use
 - Hybrid: process steps and architecture

 Benefits: close to the customer
 - Useful for telecomm because of local requirements, standards, local repair and familiarity of local infrastructure
 - Good division of labor and code ownership
  - Need clean separation of code and customized
  - Appropriate ownership
  - Need agreed handoffs

 Problems:
 - Trust - between core and custom
 - Compatible infrastructure issues - tool version, test labs, etc
 - Coordinating processes
 - Lack of documentation

 Coordination:
 - Clean split between core and customization
 - Handoff points
 - Documentation
Coordinating Teams

General Problems

Distribution of project “mass”
- Unequal distribution
- Central site is where decisions are made
  - locus of informal conversation
  - Aware before formal decisions made
- Satellite sites, constantly surprised
  - Seems like fighting upstream instead of going with the flow
- Out of sight, out of mind
- Solution: more time in travel, communication and information gathering

Finding experts
- Lack of expertise of distant system areas
- Lack of understanding of the internals of components at remote sites
- Lack of knowledge about what happens during other processes
- Lack of knowledge of core or how core is customized
- Solution: personal networks, travel and face to face time
Project Coordination

- Model of software development projects:
  - Organization
    - Defines management and organizational structure
  - Process
    - Activities, transformations, dependencies and interactions to produce software artifacts
  - Technology
    - Defines technical aspects of the artifacts and the tools applied to them

- Our hypotheses
  - Organization is not independent of process
  - Process is not independent of technology

- Two projects considered – they have
  - executed a complete development cycle
  - well-documented post mortems of their experience
  - quantitative data about interval, quality and cost.
Project Coordination

- **Project variables**
  - Cost, quality and time interval
  - Will focus on one variable: time interval

- **Reasons for focus:**
  - Reduction in interval enables faster response to customer
  - There are always inefficiencies in established processes
  - Customer-added value versus no-added value
  - Reduced interval → reduced carrying costs

- **Analysis approach:**
  - Compare established vs case study
Project Coordination

YO Packet Feature Development
- IPBG - ISDN Packet Business Group
- CNCS - Conditional Notification & Channel Selection
- PTIS - Packet Trunk Interface Standard
- MLHG - Packet Multi-Line Hunt Group

Code size, Staff size, Faults
- IPBG 12.8 9.3 3
- CNCS 19.1 13.9 4
- PTIS 14.0 10.2 2
- MLHG 8.8 10.2 2
- total 54.6 39.8 12

Challenge:
- Reduce development interval from 16 to 12 months
- Product quality goal: .23 faults/KNCSL
Project Coordination

Standard Development

- many formal handoffs
- quantized monthly intervals

Milestones and times

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Milestone Description</th>
<th>Timeframe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q10</td>
<td>FSD Complete</td>
<td></td>
</tr>
<tr>
<td>Q9</td>
<td>Requirements DS</td>
<td>3 months</td>
</tr>
<tr>
<td>Q8</td>
<td>Design DS</td>
<td>3</td>
</tr>
<tr>
<td>Q7</td>
<td>DU DS, Coding, DU Test</td>
<td>4-6</td>
</tr>
<tr>
<td>Q6</td>
<td>Capability Test Pass</td>
<td>2-3</td>
</tr>
<tr>
<td>Q5</td>
<td>Fit Test Pass</td>
<td>3</td>
</tr>
<tr>
<td>Q2</td>
<td>SV/FOA Complete</td>
<td>4-5</td>
</tr>
</tbody>
</table>
Project Coordination

YO Development

- Alters two factors
  - The many formal handoffs
  - Quantized monthly intervals

Solution

- Change from functional organization to interdisciplinary team
- Team empowered to make technical decisions

Results

- Minimized the number of formal reviews → interval reduced
- Exploited characteristics of problem → less fault insertion
  - Eg, same expert designed all the difficult fault recovery scenarios
Project Coordination

FNMS-R3 Development

- Enhanced release of ~45KNCSL in C++
- On a base of 140KNCSL
- 25 people
- 3 major features and number of minor ones
- Previous release took 16 months
  - Too unresponsive to customer needs
  - Too unstable in the field

Goal:

- Shrink overall cycle time
- Improve overall quality by removing defects early
- Decouple features from each other
  - So high priority could be delivered as early as possible
Project Coordination

Standard Development

- No formal process in place for the development of features
- Schedule-driven - ie, management plan
- Incremental development -
  - 5 major releases planned
  - But much higher due to fixing problems
- System cutover 2 months late
Project Coordination

FNMS-R3 Development

- Altered
  - Added standard quality gate techniques
    - Design reviews
    - Code inspections
  - Decoupled features that could be developed in parallel
  - Functional \rightarrow\ interdisciplinary team organization
  - Empowered teams to be responsible for their feature and feature specification
  - Within individual developments, as much in parallel as possible

- Results
  - Reduced cycle time – 25% to 12 months
  - Short features implemented and delivered very quickly
  - Defects removed earlier with very few problems
  - Increased effectiveness of development process
Project Coordination

Observations and conclusions

- Both projects displayed the same strong trends
  - Even tho in different parts of the business
  - Even tho in different kinds of software development
- Both had strongly empowered teams
- Both did what they said they were doing