CS 390 – Lecture 11

Teams continued

- Synchronize-and-Stabilize Teams
- Used by Microsoft
- Products consist of 3 or 4 sequential builds
- Small parallel teams
  - 3 to 8 developers
  - 3 to 8 testers (work one-to-one with developers)
  - The team is given the overall task specification
  - They may design the task as they wish

Synchronize-and-Stabilize Teams (2)

- Why this does not degenerate into hacker-induced chaos?
  - Daily synchronization step
  - Individual components always work together
- Rules
  - Programmers must adhere to the time for entering the code into the database for that day’s synchronization
- Analogy
  - Letting children do what they like all day… but with a 9 P.M. bedtime

Teams For Agile Processes

- Strengths of pair programming:
  - Programmers should not test their own code
    - One programmer draws up the test cases, the other tests the code
  - If one programmer leaves, the other is sufficiently knowledgeable to continue working with another pair programmer
  - An inexperienced programmer can learn from his or her more experienced team member
  - Centralized computers promote egoless programming

Open-Source Programming Teams

- Open-source projects
  - Are generally staffed by teams of unpaid volunteers
  - Who communicate asynchronously (via e-mail)
  - With no team meetings and
  - With no managers
  - There are no specifications or designs, and
  - Little or no other documentation
  - So, why have a small number of open-source projects (such as Linux and Apache) attained the highest levels of success?

Open-Source Programming Teams (2)

- Individuals volunteer to take part in an open-source project for two main reasons
  - Reason 1: For the sheer enjoyment of accomplishing a worthwhile task
    - In order to attract and keep volunteers, they have to view the project as “worthwhile” at all times
  - Reason 2: For the learning experience
The Open-Source Learning Experience

- Software professionals often join an open-source project to gain new skills
  - For a promotion, or
  - To get a better job elsewhere
- Many employers view experience with a large, successful open-source project as better than additional academic qualifications

Open-Source Programming Teams (3)

- The members of the open-source team must at all times feel that they are making a contribution
- For all these reasons, it is essential that the key individual behind an open-source project be a superb motivator
  - Otherwise, the project is doomed to inevitable failure

Open-Source Programming Teams (4)

- For a successful open-source project, the members of the core group must be top-caliber individuals with skills of the highest order
- Such top-class individuals can thrive in the unstructured environment of an open-source team

Open-Source Programming Teams (5)

- In summary, an open-source project succeeds because of
  - The nature of the target product
  - The personality of the instigator
  - The talents of the members of the core group
- The way that a successful open-source team is organized is essentially irrelevant

People Capability Maturity Model

- Best practices for managing and developing the workforce of an organization
- Each maturity level has its own KPAs
  - Level 2: Staffing, communication and coordination, training and development, work environment, performance management, coordination
  - Level 5: Continuous capability improvement, organizational performance alignment, continuous workforce innovation

People Capability Maturity Model (2)

- P–CMM is a framework for improving an organization’s processes for managing and developing its workforce
- No one specific approach to team organization is put forward
Choosing an Appropriate Team Organization

- There is no one solution to the problem of team organization
- The “correct” way depends on
  - The product
  - The outlook of the leaders of the organization
  - Previous experience with various team structures

Choosing an Appropriate Team Organization (2)

- Exceedingly little research has been done on software team organization
  - Instead, team organization has been based on research on group dynamics in general
  - Without relevant experimental results, it is hard to determine optimal team organization for a specific product

Who Should Work with Whom? Building Effective Software Project Teams

- Research study
- Objective: find the relationship between personality composition of teams and the team performance in small IS teams.
- In particular determine effects on team performance of
  - Project leader’s personality
  - Team members’ personalities
  - Heterogeneity of personalities

Previous Studies

- Case studies of only one or two teams
- Examined only one or two personality dimensions
- Done on large software teams

This Study

- 92 IS professionals from 20 business application teams in Hong Kong; both leaders and members
- Teams of 3 to 7; average of 4
- Questionnaire-based survey
- 97% response rate

Questionnaire Survey

- 1 (low) to 5 (high) Likert scale
  - Amount of work done
  - Quality of work
  - Efficiency of team operations
  - Effectiveness of user interaction
  - Frequency of schedule adherence
  - Frequency of budget adherence
- Personality type measured with Myers-Briggs Type Indicator
Myers-Briggs Type Indicator

- Four dimensional measure of personality, each with two personality styles
  - Social interaction
  - Information gathering
  - Decision making
  - Dealing with external world
- 16 possible personalities

Social Interaction Dimension

- Extrovert (E)
  - People-oriented, sociable, enjoy interacting with others
- Introvert (I)
  - Prefer to work alone, less oriented toward social interaction

Information-Gathering Dimension

- Sensing (S)
  - Prefers to seek detailed information and actual facts
- Intuitive (N)
  - Tends to make impressions without details, more imaginative and futuristic

Decision-Making Dimension

- Thinking (T)
  - Makes decisions based on logic and objective considerations
- Feeling (F)
  - Makes decisions based on subjectivity and personal considerations

Dealing with External World

- Judging (J)
  - Organized and establishes hard deadlines
- Perceiving (P)
  - Flexible and view deadlines as guidelines

Best Personality for the Job
Team Leader Personality
- Information gathering: intuitive outperformed sensing
  - Broad-based, whole-picture-oriented, innovated ability to create and assess alternate solutions
- Decision making: feeling outperformed thinking
  - People-oriented, considers effect on individuals, good for managers

System Analyst Personality
- Decision making: thinking outperformed feeling
  - Analytical skills more important than behavioral skills
  - In small teams, may need to deal with wide range of tasks

Programmer Personality
- Social interaction: extroverted outperformed introverted
  - In small teams, interact with analysts, other programmers, operators
  - Surprising to authors

Heterogeneity
- Leaders and members
  - Difference between leader and average of members
  - Higher social interaction and information gathering heterogeneity seen in more productive teams
- Among members
  - No significant impact
  - In a small team, everyone is involved with every phase

Building Better Software Teams
- Optimal personality allocation in small projects is different than in large projects
- Ensure heterogeneity between team leader and team members
- Team leaders should be intuitive and feeling
- System analysts should be thinking and sensing (to balance intuitive team leader)
- Programmers should be extroverts

Reference