People and Groups

Massimo Felici
People and Groups

• Software development is done by human beings, for human beings.

• Running successful projects requires understanding the psychology of individuals, the dynamics of small groups, and how large organisations work.

• Unfortunately, most such knowledge is fuzzy, equivocal, hard to distill into soundbites, and thus very hard to teach.

• The topics that we and others focus on are what is communicable, codifiable, etc., but they are only a small part of the total.
Human Factors in Software Development

- Modern high-level languages, libraries, and reuse now allow individuals and small teams to tackle much larger projects than before.

- Even so, there will always be some projects that require large teams, and these still work (badly) as they always have.

- Processes will be successful only to the extent that they take into account how people and teams actually behave.

[Alistair Cockburn, Growth of human factors in application development, Humans and Technology Technical Report HaT.TR.95.04.]
Team Roles

• Popular way of understanding how different personalities behave on a team

• Roles: Plant, Resource Investigator, Co-ordinator, Shaper, Monitor Evaluator, Teamworker, Implementer, Completer-Finisher, Specialist

• Good teams have a good mix of personalities

• People have different roles on different teams and at different times

• Overly commercialised, but basic idea is reasonable

“A tendency to behave, contribute and interrelate with others in a particular way.” [http://www.belbin.com/]
Small-Group Development

Tuckmans (1965; 1977) summary of how small groups change over time has been very influential:

- **Forming**: orientation, testing and dependence
- **Storming**: resistance to group influence and task requirements
- **Norming**: openness to other group members
- **Performing**: constructive action
- **Adjourning**: disengagement

Regardless of whether these stages really apply to specific projects, they have been useful at least in getting people to think about how groups behave.
Slide 4: References


http://dx.doi.org/10.1177/105960117700200404
Large Organisations

- Bureaucratic organisations like governments, universities, and large companies have a peculiar logic all their own.

- Everything is done by individuals, yet to be manageable the organisation needs to ensure consistency, repeatability, predictability.

- Various standards have been proposed to reach those ends: CMM (Capability Maturity Model), ISO-9000/9001, numerous IEEE standards, etc. — Nearly all focus on the process, not the end product, which allows them to be general (but may miss the point!).
Capability Maturity Model

Originally developed for US military subcontractors, as a way to make distinctions between them. Identifies five levels of process maturity for an organisation:

1. **Initial** (chaotic, ad hoc, heroic) starting point for use of a new process
2. **Repeatable** (project management, process discipline) process is used repeatedly
3. **Defined** (institutionalised) process defined/confirmed as standard part of business
4. **Managed** (quantified) process management, measurement takes place
5. **Optimising** (process improvement) process management includes deliberate process optimisation/improvement
Capability Maturity Model

CMM Pros:

• Allows large, bureaucratic organisations to make overall judgements about a subcontractor’s ability to work well with a large, bureaucratic organisation

CMM Cons:

• May simply favour process-heavy organisations over innovative ones (by design) or those responding quickly and flexibly to market pressures

• Not much empirical data on how well CMM correlates with some more tangible measure of success

• CMM compliance is self-reported
Capability Maturity Model

• Agile processes like XP are well defined processes, as CMM requires

• Agility conflicts a bit with the type of heavy documentation required by CMM

• Still, it is possible to have an agile process with a high CMM level

[For more details on CMM, see: http://www.sei.cmu.edu/cmmi/; and Humphrey, W. S. (1989), Managing the Software Process, Addison-Wesley.]
ISO 9000/9001 standard

Originally from manufacturing, specifically as a way to evaluate UK government munitions contractors. Contains:

- set of procedures covering all key processes in the business
- monitoring processes to ensure they are effective
- keeping adequate records
- checking output for defects, with appropriate corrective action where necessary
- regularly reviewing individual processes and the quality system itself for effectiveness
- facilitating continual improvement
Root Cause Analysis

- CMM, ISO-9001, and other ‘meta’ processes often focus on introspection and postmortem analysis.

- When a project completes or reaches a significant milestone (perhaps even for every iteration), it is an opportunity to understand what went right and wrong, with relatively little work.

- CMM and ISO-9001 focus on applying the lessons learned, so that successful approaches can be applied widely, and unsuccessful ones avoided.

- The key is to find the root cause, i.e. the deeper, underlying reason why something went wrong (or right!).
Root Cause Analysis

Five Whys — common technique for doing root cause analysis — if the problem is “My car will not start”, multiple questions get at the underlying cause:

1. Why? — The battery is dead.

2. Why? — The alternator is not functioning.

3. Why? — The alternator belt has broken.

4. Why? — The alternator belt was well beyond its useful service life and has never been replaced.

5. Why? — I have not been maintaining my car according to the recommended service schedule. (root cause)
Summary

• Understanding how individuals, groups, and bureaucracies work is crucial for running successful projects

• Difficult to achieve by book learning; project leaders need to be students of human nature

• Try not to bludgeon the humanity out of your people with heavy-handed processes

• Yet somehow you need to make results ok for the organisation
Required Readings

  
  http://dx.doi.org/10.1109/52.965798