People and Groups

Software Architecture, Process, and Management

Technical prowess will have some bearing on the success or failure of your project. But the management of a project is largely the management of people.

Human Factors

- 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

Belbin's Team Roles 1/3

- *"A tendency to behave, contribute and interrelate with others in a particular way."*
  - Plant: Creative, unorthodox and generators of ideas
  - Resource Investigator: has their finger firmly on the pulse of the outside world
  - Co-ordinator: recognise abilities in others and very good at delegating tasks to the right person for the job
  - Shaper: winning is the name of the game, pursues objectives with single-mindedness
  - Monitor Evaluator: fair and logical observers, can often detach themselves from bias, good whenever the team needs a deciding vote

Belbin's Team Roles 2/3

- *"A tendency to behave, contribute and interrelate with others in a particular way."*
  - Teamworker: oil between the cogs, often a peacemaker, can go unnoticed until not there
  - Implementer: Turns suggestions and ideas into positive action
  - Completer Finisher: Perfectionist, can be relied upon to have performed tests etc.
  - Specialist: Passionate about learning their field, will often be the authoritative source, and will go and seek out information they do not know

Belbin's Team Roles 3/3

- *"A tendency to behave, contribute and interrelate with others in a particular way."*
  - Popular way of understanding how different personalities behave on a team
  - 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

The Peter Principle

- States that *"in time, every post tends to be occupied by an employee who is incompetent to carry out its duties"*
- Well performing employees at one-level get promoted until they are not performing well

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, 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!)

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's 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!)

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
     - This is root cause
     - And a process that we might change for the future

Hypothetical Schedule

- But all this makes a few assumptions:
  - The tasks are *perfectly divisible*
  - Additional team members do not slow down current team members
  - Training new developers on the work done so far is either not required or takes effectively zero time

Metcalfe's Law

- As the team grows, so too does the number of communication channels
- This increases the communication overhead and time lost to communication
- I spoke before about the effects of interrupting a programmer
- A programmer takes between 10-15 minutes to start editing code after resuming work from an interruption.
- When interrupted during an edit of a method, only 10% of times did a programmer resume work in less than a minute

The Internet

- The Internet has made communication simply easier
- Emails do not necessarily disrupt concentration in the way that a phone call may
- Reply-all has given some degree of ammunition against Metcalfe's law

Internet 2-way/Broadcast

- It's not the fact that we can send to each other, we already had ways to do that
- It's not the fact that we could easily send to many people, we already had ways to do that
- But "Reply-all" was a social function that meant we could form groups, because anyone receiving the broadcast message could respond to the entire group
  - Letters to the editor is the closest thing we've had before
  - But that is inherently moderated by the producer
- However if this is a major contributing factor in the reduction of software project failures then one might expect that we've come as far as we're going to come based on that alone

Group Behaviour

- So communication overheads can mean that adding members to a group can slow down those original members
- Unfortunately it gets much worse; we will now look at the behaviour of groups and how members can affect other team members

Herbert Kelman's Three Kinds of Conformity

- Compliance is public conformity, while possibly keeping one's own original beliefs for yourself
  - Compliance is motivated by the need for approval and the fear of being rejected
- Identification is conforming to someone who is liked and respected, such as a celebrity or a favourite uncle
  - This can be motivated by the attractiveness of the source, and this is a deeper type of conformism than compliance
- Internalisation is accepting the belief or behaviour and conforming both publicly and privately, if the source is credible
  - It is the deepest influence on people and it will affect them for a long time
  - Can blind people in, for example, language wars

Group Conformity and Software Engineering

- Following the crowd can mean that people will not stand up and say something when they see a problem
- Sometimes the entire group believes there is a problem
- But unfortunately they all believe they are the only one to see it as a problem
  - Often described as *"no one wants to be the first to complain/demoralise/blame"*
  - But it's more, no one wants to be the **only** one
- We have already seen an application against this kind of behaviour
  - Recall planning poker insists that each estimator makes their guess without knowing the estimates of the others
  - This is specifically to prevent the first estimate influencing the others in the group

Bad Apples

- Less so today in the world of constant harvest and global export markets, but it wasn't so long ago people would harvest apples in Autumn and store them in barrels for consumption over the winter
- A single apple that is bad, generally because it has been bruised or perhaps eaten by something small, gives off ethylene
- Unfortunately, ethylene causes good apples to ripen quicker
- Hence a single bad apple can cause the entire barrel to be wasted before they can be eaten
- For this reason apples were stored in flat boxes and one of the chores of the day was to go and check the boxes for any bad apples and **remove them**

Bad Apples in Groups

- This is of course where our term “bad apple” meaning someone in group comes from
- Though often it seems today the expression has lost something of its original meaning and no longer emphasises the damage to a group that a single *“bad apple”* can do
- The theory is often that people follow along with whatever the general behaviour of the group is
- Will Felps has performed some experimentation on the surprisingly powerful effect of bad apples

    Which is in itself an important observation

Groups of Bad Apples

- Will had college students put in groups of four
- Each group was given the task of making some basic management decisions within 45 minutes
- As motivation, the best team is rewarded with $100 each
- Link to the research **article**

The Withholder of Effort

Triggers

- Perception of a *withholder of effort* among the rest of the team triggers several undesirable responses:
  - Social comparison, in which people compare their own situation to others
  - Feeling under-rewarded produces stronger psychological effects than feelings of over-reward
  - Feelings of inequity are stressful and result in team members wishing to restore equity, which can be easily achieved by withholding effort themselves
  - In doing so can risk destroying the otherwise favourable relationships they already maintain with other members of the team

Removing Bad Apples

- Rather than **adding** new team members, a possible course of action is to remove the worst performing member of the team
- They can simply be moved to a separate project
- In a study of 32 management teams, Larson and LaFasto found that the most consistent *and intense* complaint from team members was that their team leaders were unwilling to confront problems associated with poor performance by individual team members
- see "TeamWork: What Must Go Right/What Can Go Wrong" Larson and LaFasto, 1989
- To do this, you must identify any *bad apples*

Steve McConnell's bad apple traits

Does/did anyone on your team?

- Cover up their ignorance/bluff rather than learn from teammates:
  - "I don't know how to explain my design; I just know it works"
  - "My code is too complicated to test"
- Have an excessive desire for privacy:
  - "My code does not need any review"
- Act territorial:
  - "No one else can fix bugs in my code, I'm too busy to fix them right now but I'll get back to them next week"

Steve McConnell's bad apple traits

Does/did anyone on your team?

- Grumble about team decisions and continue to revisit old discussions long after the team has moved on:
  - "We should have used Java instead of C# like I said"
  - "This would be easy in .net"
- Receive wisecracks/jokes about the same person regularly, software developers often won't complain directly:
  - "Oh you're going over the plan with Jimmy? good luck!"
- They don't pitch in on team activities, even those not directly related with work, such as post-release drinks
  - "Can I have Tuesday off, I know it's two days before a release, but **my** code is finished and tested"

Striking a balance

- As a project manager one must strike a balance
- We do not want to allow group conformity to prevent problems being aired, by either a single developer or worse by everyone who is silently conforming
- At the same time, one does not want a JavaScript Joe to hamper the team
- In English we have the expression:
  - *"The squeaky wheel gets the oil"*
- In Japanese, they say it differently:
  - *"The stuck out nail is hit with the hammer"*