

## **Managers: What are they good for?**

In addition to the presentation this session contained two exercises for the audience. First the audience was divided in two, managers on the left of the room and developers on the right. They were then asked to form small groups of 4-5 people each.

For the first exercise each group was asked to list things they, as managers or developers, didn't like about the other. After 10 minutes the teams were stopped and as facilitator I made one list of all the developers concerns on a flip chart at the front of the room. Next I made another list of all the manager complaints written up by myself as facilitator.

There followed a short discussion on the two lists, similarities, differences and contradictions. I then gave a short presentation before the second exercise.

In the second exercise the managers were asked to make a list of things they felt they should spend their time doing and developers asked to make a list of things they thought their managers should be doing. This time only the developer lists was compiled into a single public lists, items which also appeared on the managers list were highlighted.

Again there was some discussion but by now time was closing in. I skipped most of the rest of the presentation in order to close the session on time.

The three lists which follow are taken from those interactive sessions with minimal editing.

### ***Things Managers don't like about developers***

- Developers who only bring problems not solutions
- Developers who lack focus and need hand holding
- Developers who treat their managers as personal assistants
- Developers who are territorial about their code and not team oriented
- *Glamour coding*: doing what's cool, not what's important
  - Lack of care for the final result and the stakeholders
- Developers who won't admit they don't know what to do
- Developers who won't admit they have a problem
- Egotistical developers
- Developers who fear their managers and assume they are trying to trick the them
- Poor time estimates
- Unfounded optimism
- Developers don't see the shit from above; managers have to deal with a lot too

### ***Things developers don't like about managers***

- Failure to take personalities into account

- Treating developers like 'plug compatible' interchangeable resources
- Managers who expect definite answers
- Managers who don't trust developers
- Managers who don't take time to start a project properly
- Managers who change priorities and scope on the fly
  - Who encourage scope creep
  - Who change tasks when developers are half way through
  - Who say Yes to customers too easily
- There are very few good managers
- Managers are not interested in the detail
- Managers who don't control budgets or won't spend money
- Managers exercise decision power without responsibility
- Managers who hold secrets as power
- Managers who don't come to the pub for a drink with the team
- Managers get paid more than developers
- Managers who are absent when needed; and managers who are constantly badgering for results and news
- Egotistical managers
- Managers who are too positive
- Managers who like buzzwords
- Managers who are addicted to processes and reports
- Managers who are unrealistic and do not compromise
- Managers who filter too much information
- Managers who ask for problems before cures
- Managers who leave at 5pm
- Lack of respect for developers
- Lack of trust of developers
- Managers allocate tasks but don't allow time for learning to occur
- Managers who don't consult developers
- Managers who don't learn
- Managers promoted on the Peter Principle: *Everyone is promoted to the point where they are incompetent*

### ***What developers would like from their managers***

Items marked with an asterix\* were also listed by managers on their lists.

- Help developers focus on their job\*

- Shield developers from others
- Filter information
- Be a communicator
  - Act as a go-between between the outside world and the development team
  - Inside-out and outside-in communication
  - Translate management speak and business objective to something understandable
  - Explain pressures within the business
  - Share praise for the team
- Value developers\*
  - Build happy teams and happy developers\*
  - Consider the needs and interests of developers
  - Understand people
- Sack some people\*
  - Work to consensus not with authority
  - Act as an arbitrator between people\*
  - Represent and champion developers elsewhere in the company
- Earn developers trust and work closely with them
- Help developers grow and learn\*
- Provide feedback on actions\*
- Mediate personalities\*
- Understand the unpredictable nature of development work
- Limit work and plan on a sensible timescale
- Be open\*
- Support the team at all times
- Stay up to date with development progress
- Act faster
- Listen to developers more
- Understand technical debt
- Leave developers alone
- Manage different personalities with different management styles

## ***Conclusion***

While each list contains some contradictions of its own these are to be expected given that different people have different experiences, and that several sub-groups contributed to each list. However, when the initial lists

were drawn up most of the items listed were supported by more than one sub-group.

Running through the first two lists seems to be a general lack of trust between developers and managers. Many of the same issues are cited, or the same issue is cited seem from different perspectives, for example, both sides consider the other to be egotistical and over optimistic.

However, the things that stand out are the similarities not the differences. This is particularly true in the third list where developers listed what they wanted their managers to do. The core issues also appeared on the managers lists.

Perhaps what is surprising is that developers want their managers to act faster and even fire people who are under performing. It seems the developers have a harder view of the world than the managers.

For managers the overall lesson seems to be that they need to build trust with their developers. The first step to this would be clear and open communication. Developers want shielding from elsewhere in the company and management chain but they resent information filtering. Getting the balance right between *shielding* and excessive *filtering* is a difficult line to walk. To further complicating factors managers say developers do not appreciate that which they do.

It is also true to say that managers need to find out what their own teams and developers want. Different people, and different teams, need managing in different ways. It is wrong to assume that a *one size fits all* management style will work in all environments. Managers need to be adaptable.

### ***See also***

- Original conference presentation at <http://www.allankelly.net/static/presentations/accu2008/Managers.pdf>
- Presentation handout at <http://www.allankelly.net/static/presentations/accu2008/ManagersHandout.pdf>