

Notes - Organizational Development - Teams - Part 2

- Dr Nick Hayward

A brief introduction to team management and structure with example usage for software projects.

Contents

- Intro
- Team control and metrics
- Team communication
 - email usage
 - conference and meetings
- Resources

Intro

Whilst some developers may prefer to work as individuals, crafting their own solutions to a given application or structural problem, there are notable benefits to effective collaboration.

Team management, organisation, and structure can provide opportunities to work effectively, achieve tight schedules, and diversify talents across a broad range of developers, managers, and personnel. There's no magic formula, but we may review various goals and reasons to help create an environment to encourage a successful team.

We often define required specifications for software development, such as a *software requirements specification* (SRS) or *software design document* (SDD). We may also consider and outline such requirements and expectations as we initially form a team. These might include, for example, what we need for the team, responsibilities, expected output, size and scope, &c.

As part of the overall department or company structure, we may also consider a team as a module or component that is required to produce a defined output, meeting set requirements, passing various tests, which is then passed as the input to another process, system or team.

A well functioning team is commonly able to work independently to solve problems, whilst maintaining awareness, where appropriate, of the larger internal company structure. As such, successful teams will usually be granted a semblance of independence and responsibility to creatively solve defined problems.

Team control and metrics

A consideration of metrics may vary from team to team or, perhaps, department by department.

However, even a basic set of metrics can help each team, and its leader, monitor and review internal performance and success. Such metrics should have intrinsic value, providing a comprehensive view of progress, contributing to the overall report commonly prepared to inform management.

A common benefit of underlying data for such reviews and reports is the reduction in potential for emotional ambiguity when reviewing the success of a team. Within a given department, such reports may also be the result of teams competing for success, rewards, internal bonuses &c. However, whilst this may be perceived as beneficial for pure end results, there is the danger of poorly balanced reviews between disparately focused teams, effectively creating unfair comparisons.

With metrics in mind, we may instead prefer to compare like-minded teams, which may include earlier or projected versions of the current team. Such metrics, therefore, might include the following

- issues, features - open and closed, rate of completion and implementation
- feature releases - frequency, quantity
- performance - including load metrics for data, server &c. where appropriate
- sprints - performance and progress per goal, sprint

- balance between rapid delivery and perceived code excellence - e.g. allocation of resources per sprint cycle to refactoring code, fixing bugs &c.
 - perception of required technical debt...
 - e.g. 15 to 20% of resources might be allocated to keeping technical debt under control, thereby helping to control and maintain a high level of software quality
- time spent or required to address support issues
- ...

Such metrics should not become a team role, in and of itself, but part of team analysis and collation of performance data. These tools have become a popular and useful part of the way developers and teams use sites such as GitHub and GitLab.

Another decision relative to metrics is a consideration of when, and how often, such metrics may need to be collected by the team's members and its leader. Daily, weekly can provide useful insights to help prevent issues accumulating and persisting for too long. However, a balance will need to be struck between development performance and review without reducing productivity.

Such decisions will need to be made by the team leader and department managers.

Team communication

A key part of team communication, and good leadership, can be the effective balance between transparency and respect for appropriate confidentiality with team, department, and company matters.

Transparency will often help foster a collaborative, comfortable team and development structure. However, this should not be confused with immediate release and publication of all confidential or project details the moment they are learnt by a team member or leader.

This balance may seem obvious, but there is a time and a place to share information, in particular appropriate to team progress, development, and well being.

A team or project leader, perhaps at the department level, often acts as a filter or conduit for broader company level information and updates. Some of these notifications need to remain confidential until an appropriate time.

In effect, such transparency is often best conveyed in a thoughtful, considerate manner to help effectively communicate and articulate the vision and goals for the overall company and management.

To help with such communication, there are various mediums we may use on a regular basis, which require consideration.

email usage Whilst chat and instant messaging options, including *Slack* and *Teams*, have become increasingly popular in recent years, in particular amongst software development projects, email persists as a common communication tool.

A noted benefit for email is the ability to differentiate the underlying nature of the communication from real-time, almost transient in nature for chat options compared with the ability to delay, schedule, and create a historical record for many email based conversations.

Email may be perceived, still for many situations, as preferable for broadcasting perceived critical, widespread team, department, and company announcements. A good way to adopt this differentiation between real-time options and email is to create and maintain specific aliases per group. This alias, of course, should, initially, only include the specific team members and collaborators.

Maintenance of such lists may also form a useful part of introductory, onboard processes and similar exit or offboard situations.

A few sample topics commonly addressed with email communications might include the following,

- company announcements
- group wide announcements

- new hires and additions to the team, department &c.
- departures and team personnel updates
 - might also include restructuring announcements for team roles &c.
- appropriate general news and items of interest
 - might be team, department or company wide in nature...

A key aspect of productive email usage is to ensure recipients become attuned and aware of the general nature and tone of such messages. In effect, avoid email communications for irrelevant, informal topics, which might be ignored by recipients. The content of email messages should be read by all, and not simply dismissed upon receipt.

Many companies may also maintain a permanent record or store for the email messages sent and received by its employees. As soon as a message is sent, for example, it may be persisted as part of a company's ongoing record of business. This may be necessary for legal reasons or simply as part of the underlying technology infrastructure, often provided by services such as Microsoft's Office. Such persistent stores are designed to resist manipulation, providing an effective solution for audit and legal requirements.

conference and meetings To keep a project, department, and company running smoothly, various meetings will be essential. You need to maintain a personal bond with your group's members, as well as other collaborative team leaders. We might consider the following example meeting types as common examples for project management and leadership.

- *team and group sessions*
 - regular meetings and updates amongst team members, and project leads, helps promote collaboration and information sharing beneficial for both the immediate project and associated groups
 - usually held once a week, with a clear focus on various perceived *high-level* issues currently impacting the project team
 - clearly define agenda items, order, and time per item with time reserved at the end of the meeting for broader discussion, comments &c.
 - important to ensure minutiae does not derail the meeting - instead, make a note of such specific details and items, and move them to a separate specific meeting or review
 - such meetings should be focused, less than one hour where possible
 - permit each agenda item to receive fair attention, allowing group leaders to report with equal relevancy and time constraints
- *development and engineering sessions*
 - such meetings provide a clear opportunity to outline and discuss the current state of an application or product
 - usually scheduled for each week or fortnight, including rapid sessions highlighting general overarching considerations and status
 - common topics for consideration might include major release cycles, client gains and losses, general structural improvements for development processes, &c.
 - shorter sessions, perhaps only half an hour, avoid potential for minutiae in code details, specific bugs, issues &c. - these have their own tools and methods for discussion, resolution...

Other common types of meetings might include *ad-hoc* scheduled items, broader *town halls*, and even one on one sessions.

However, be careful how you use meetings to organise and manage a group. Whilst beneficial for structure, resolution, and general collaboration, they can also impact the productive life of a group, department or company if they are not structured and executed in a clear, concise manner.

A few rules to consider for meetings in general,

- distribute an agenda before the meeting begins with sufficient time for review, including associated reading or reference material
- meetings often benefit from a simple consideration of appropriate attendance
 - e.g. those who will or need to contribute should be involved

- not always necessary to include those who will be *in the know*...
- potential for interruptions should be removed
 - e.g. phones &c. should be switched off
 - laptops might be considered solely for presentations, and not necessarily open as a potential source of distraction with email, instant messages &c.
 - catering is fun for social events, but a business meeting should promote focus and concentration
 - i.e. does the meeting really need to be long enough to merit catering
- ...

Resources

- “13 ways you can make meetings more effective” - Forbes - <https://www.forbes.com/sites/forbesbusinesscouncil/2019/10/ways-you-can-make-meetings-more-effective/>
- Herold, Cameron. “Meetings Suck: Turning One of The Most Loathed Elements of Business into One of the Most Valuable”. 2016. Lioncrest Publishing.
- “Facilitating Meaningful and Effective Meetings: Remotely and Beyond” - Smart Tech - <https://www.smarttech.com/en/education/resources/article/effective-meetings>
- “Strategic Meetings For Learning: Merging Business Discussions And Skill Enhancement” - Tech Journal - <https://techjournal.org/strategic-meetings-business-and-skill-enhancement>
- “What it takes to run a great hybrid meeting” - Harvard Business Review - <https://hbr.org/2021/06/what-it-takes-to-run-a-great-hybrid-meeting>
- “What is an effective meeting?” - McKinsey & Company - <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-an-effective-meeting>