Teamwork, Leadership and Motivation

Lecture 7 by Professor Vladimir Geroimenko

Module “Software Project Management“

06 November 2016 - Teaching Week 7

Textbook reference: Chapter 12

Lecture Outline

1. The nature of teams and teamwork
2. Basic stages of team development
3. Team roles
4. Delphi approach
5. Teamwork techniques for software development
    - Egoless programming
    - Chief programmer teams
    - XP
    - Scrum
6. Virtual Teams
7. Time/place constraints on communication
8. Communications plans
9. Managing versus Leading
10. Motivation theories

Individuals, Teams and Teamwork

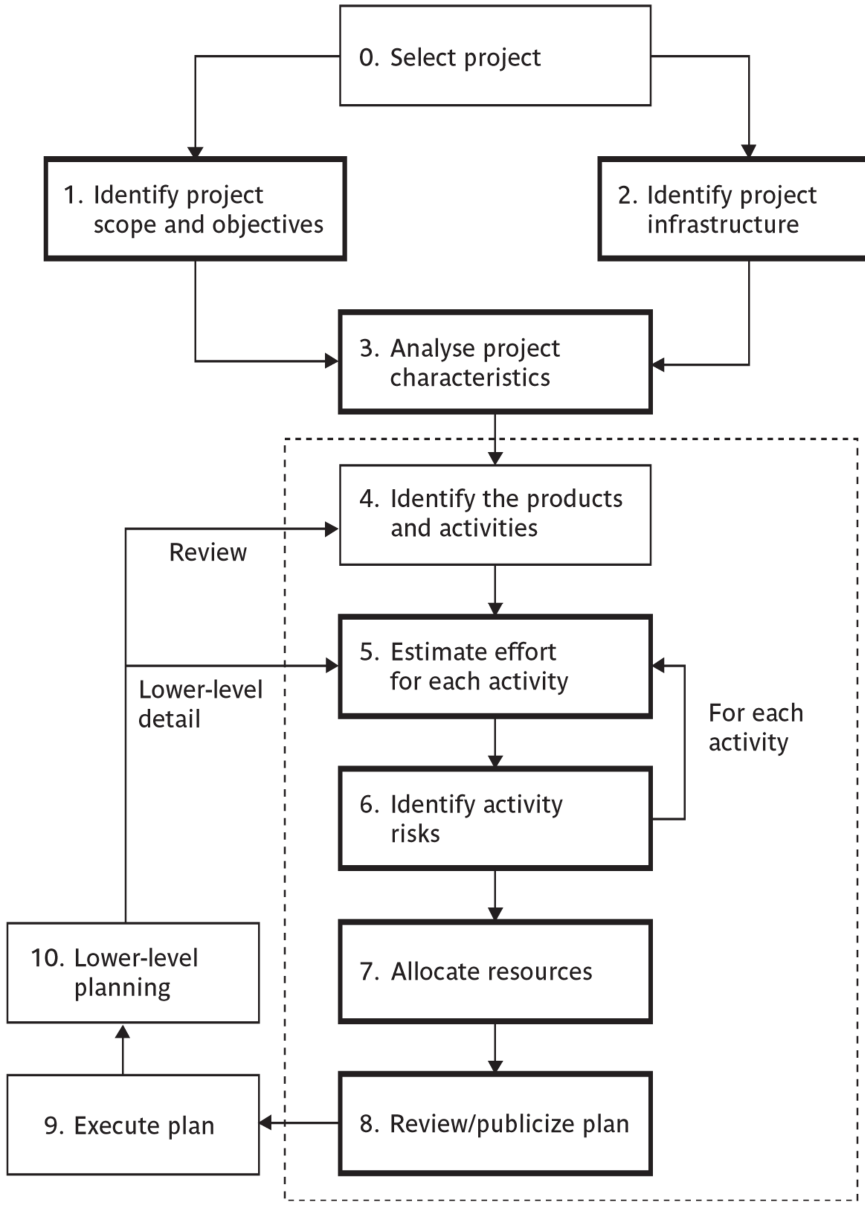
A **team** is a group of individuals working in cooperative manner toward *common, shared goals*.

Please note:

- team members are individuals with individual goals, agendas, motivations, desires, and attitudes
- a team has a shared vision and shared work products
- team members are willing to help one another
- a vision of common, shared goals seldom happens spontaneously

Factors That Contribute to Teams / Teamwork

- Appropriate number of people with correct skill mixture
- Respect for other team members
- Respect for managers and leaders
- Willingness to be team members
- Shared ownership of the work products
- Good communication skills
- Good communication channels
- Good working environment
- Having some fun together


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/publicize plan

Some places in the Step Wise framework influenced by collaborative working

Five basic stages of team development

1. Forming
2. Storming
3. Norming
4. Performing
5. Adjourning

Forming: team get to know each other
Storming: conflicts arise; power and operation methods
Norming: conflicts settle, and group identity emerges
Performing: focus on tasks
Adjourning: team deliver and disband

Becoming a Team

Teams need a balance of different types of people:

Chair: calm, strong, and tolerant

Plant: very good at generating ideas and solutions

Monitor: good at evaluating ideas.

Shaper: keeps team focused at important issues (redirects attention)

Team worker: keeps the teams spirits

Resource investigator: good at finding resources: physical and info

Finisher: good at completing tasks

Company worker: willing to take the less attractive tasks

More team roles (1)

- **The co-ordinator** – good at chairing meetings
- **The ‘plant’** – an idea generator
- **The monitor-evaluator** – good at evaluating ideas
- **The shaper** – helps direct team’s efforts
- **The team worker** – skilled at creating a good working environment

More team roles (2)

- **The resource investigator** – adept at finding resources, including information
- **The completer-finisher** – concerned with getting tasks completed
- **The implementer** – a good team player who is willing to undertake less attractive tasks if they are needed for team success
- **The specialist** – the 'techie' who likes to acquire knowledge for its own sake

Please note: A person can have elements of more than one type. 30% of people can not be classified at all.

Group performance

Some tasks are better carried out collectively while other tasks are better delegated to individuals

- *Additive tasks* – the effort of each participant is summed
- *Compensatory tasks* – the judgements of individual group members are summed – errors of some compensated for by judgements of others
- *Disjunctive tasks* – there is only one correct answer
  - Someone must come up with right answer and persuade the others
- *Conjunctive tasks* – the task is only finished when all components have been completed

Barriers to good team decisions

- Inter-personal conflicts
  - Conflicts tend to be a dampened by emergence of *group norms* – shared group opinions and attitudes
- *Risky shift* – people in groups are more likely to make risky decisions than they would as individuals

Delphi approach

To avoid dominant personalities intruding the following approach is adopted:

1. Enlist co-operation of experts
2. Moderator presents experts with problem
3. Experts send in their recommendations to the moderator
4. Recommendations are collated and circulated to all experts
5. Experts comment on ideas of others and modify their own recommendation if so proposed by the others
6. If moderator detects a consensus, stop; else back to 4

Team ‘heedfulness’ (heedful = careful, helpful)

- Where group members are aware of the activities of other members that contribute to overall group success
- Impression of a ‘collective mind’
- Some techniques to promote this:
    - Egoless programming
    - Chief programmer teams
    - XP
    - Scrum

Egoless programming

- Gerry Weinberg noted a tendency for programmers to be protective of their code and to resist perceived criticisms by others of the code
- Encouraged programmers to read each others code
- Argued that software should become communal, not personal – hence ‘egoless programming’

Chief programmer teams

- Fred Brooks was concerned about the need to maintain ‘design consistency’ in large software systems
- Appointment of **chief programmers**, with responsibilities for defining requirements, designing, writing and test software code
- Assisted by a support team: **co-pilot** – shared coding, **editor** who made typed in new or changed code, **program clerk** who wrote and maintained documentation and **tester**
- Problem – finding staff capable of the chief programmer role

Extreme programming

XP can be seen as an attempt to improve team heedfulness and reduce the length of *communication paths* (the time between something being recorded and it being used)

- Software code enhanced to be self-documenting
- Software regularly refactored to clarify its structure
- Test cases/expected results created *before* coding – acts as a supplementary specification
- *Programming in pairs* – a development of the co-pilot concept

Scrum

- Named as an analogy to a rugby scrum – all pushing together
- Originally designed for new product development where ‘time-to-market’ is important
- ‘Sprints’ increments of typically one to four weeks
- Daily ‘scrums’ – daily stand-up meetings of about 15 minutes
- Unlike XP, requirements are frozen during a sprint
- At the beginning of the sprint there is a sprint planning meeting where requirements are prioritized
- At end of sprint, a review meeting where work is reviewed and requirements may be changed or added to

Co-ordination of dependencies (1)

What sort of communications are needed between a team and other units?

Co-ordination theory has identified the following types of coordination:

- *Shared resources. e.g.* where several projects need the services of scarce technical experts for certain parts of the project.
- *Producer-customer (‘right time’) relationships.* A project activity may depend on a product being delivered first.
- *Task-subtask dependencies.* In order to complete a task a sequence of subtasks have to be carried out.

Co-ordination of dependencies (2)

- *Accessibility ('right place') dependencies.* This type of dependency is of more relevance to activities that require movement over a large geographical area, but arranging the delivery and installation of IT equipment might be identified as such.
- *Usability ('right thing') dependencies.* Broader concern than the design of user interfaces: relates to the general question of *fitness for purpose,* e.g. the satisfaction of business requirements.
- *Fit requirements.* This is ensuring that different system components work together effectively.

Virtual Teams

- When the team is not physically located in one office. But possibly working from home or from any dispersed spaces.
- It is sometimes needed to reduce costs, because it is difficult to allocate the needed space of a productive environment, to give programmers flexibility.

Virtual Teams - Advantages

- Can use staff from developing countries – lower costs
- Can use short term contracts:
  - Reduction in overheads related to use of premises
  - Reduction in staff costs, training, holidays, pensions etc.
- Can use specialist staff for specific jobs
- Productivity of home workers can be higher – fewer distractions
- Can take advantage of time zone differences e.g. overnight system testing

Virtual Teams - Some challenges

- Work requirements have to be carefully specified
- Procedures need to be formally documented
- Co-ordination can be difficult
- Payment methods need to be modified – piece-rates or fixed price, rather then day-rates
- Possible lack of trust when there is no face-to-face contact
- Different time zones can cause communication and co-ordination problems

Time/place constraints on communication

	Same place	Different place
Same time	Meetings Interviews	Telephone Instant messaging
Different times	Notice boards Pigeon-holes	Email Voicemail Documents

Best methods of communication (1)

- Early stages
  - Need to build trust
  - Establishing context
  - Making important ‘global’ decisions
  - ***Same time/ same place***
- Intermediate stages
  - Often involves the parallel detailed design of components
  - Need for clarification of interfaces etc
  - ***Same time/different place***

	Same place	**Different place**
Same time	Meetings Interviews	Telephone Instant messaging
Different times	Notice boards Pigeon-holes	Email Voicemail Documents

Best methods of communication by stages (2)

	Same place	Different place
Same time	Meetings Interviews	Telephone Instant messaging
Different times	Notice boards Pigeon-holes	Email Voicemail Documents

- Implementation stages
  - Design is relatively clear
  - Domain and context familiar
  - Small amounts of operational data need to be exchanged
  - ***Different time/different place communications***

- Please note: Face to face co-ordination meetings – the 'heartbeat' of the project

Communications plans

- Choosing the right communication methods is crucial in a project
- A good idea to create a **communication plan**
- **Stages** of creating a communication plan
  - Identify all the major stakeholders for the project
  - Create a plan for the project
  - Identify stakeholder and communication needs for each stage of the project
  - Document in a communication plan

Content of a communication plan

- ***What.*** The name of a particular communication event, e.g, ‘kick-off meeting’, or channel, e.g. ‘project intranet site’.
- ***Who/target.*** The target audience for the communication.
- ***Purpose.*** What the communication is to achieve.
- ***When/frequency.*** If the communication is by means of a single event, then a date can be supplied. If the event is a recurring one, such as a progress meeting then the frequency should be indicated.
- ***Type/method.*** The nature of the communication, e.g., a meeting or a distributed document.
- ***Responsibility.*** The person who initiates the communication.

Managing versus Leading (1)

- **Managing** is concerned with:
  - making plans and estimates,
  - collecting and analyzing project and product data,
  - reporting progress,
  - controlling the development process and the products,
  - identifying and mitigating risk factors.

- **Leading** is concerned with:
  - communicating with your project personnel and other stakeholders,
  - coordinating the work activities,
  - maintaining morale.

Managing versus Leading (2)

- Good managers are not necessarily good leaders, and good leaders are not necessarily good managers
  - Managing is an **analytical activity** whereas leading involves **human relations**
  - Different personality traits and different skill sets are required for managing and for leading

Please note:

Some excellent managers are poor leaders and

some excellent leaders are poor managers

Division of Responsibilities

- **The project manager** is responsible for delivering an acceptable product on schedule and within budget

- **The technical leader** (software architect) is responsible for <u>leading the project team</u> to achieve the “acceptable product”, within the constraints of schedule and budget

Please note:

On a small project, one person may play the roles

of both project manager and technical leader

Some Attributes of a Good Leader

- Listens carefully
- Delegates authority
- Facilitates teamwork
- Speaks with individuals on a daily basis
- Helps employees develop career plans

  and achieve their professional objectives

- Accept responsibility
- Facilitates communication
- Coordinates work activities
- Says "thank you" when warranted
- Resolves conflicts
- Reconciles differences
- Maintains enthusiasm
- Coaches and trains

Leadership: Position power

Position power

- *Coercive power* – able to threaten punishment
- *Connection power* – have access to those who do have power
- *Legitimate power* – based on a person's title conferring a special status
- *Reward power* – able to reward those who comply

Leadership: Personal Power

Personal power

- *Expert power:* holder can carry out specialist tasks that are in demand
- *Information power*: holder has access to needed information
- *Referent power*: based on personal attractiveness or charisma

Leadership styles

		decision-making	
		autocrat	democrat
implementation	directive	makes decisions alone close supervision of implementation	makes decisions participatively close supervision of implementation
	permissive	makes decisions alone subordinates have latitude in implementation	makes decisions participatively subordinates have latitude in implementation

Leadership styles

- **Task orientation** – focus on the work in hand
- **People orientation** – focus on relationships

- Where there is uncertainty about the way job is to be done or staff are inexperienced they welcome task oriented supervision
- Uncertainty is reduced – people orientation more important
- Risk that with reduction of uncertainty, managers have time on their hands and become more task oriented (interfering)

Mental obstacles to good decision making

- Faulty heuristics: rules of thump might be wrong
- Escalation of commitment: committing to a decision even when it is apparent that it is wrong
- Information overload: can’t see the wood for the trees

What is Motivation?

Internal and external factors

that stimulate desire and energy in people

to be continually interested and committed to a job, role or subject,

or to make an effort to attain a goal.

What motivates software developers?

- A workshop report in 2010
- Available at http://accu.org/index.php/journals/1703

Motivation theories

- Taylor
- Maslow
- Herzberg
- Vroom
- Hackman

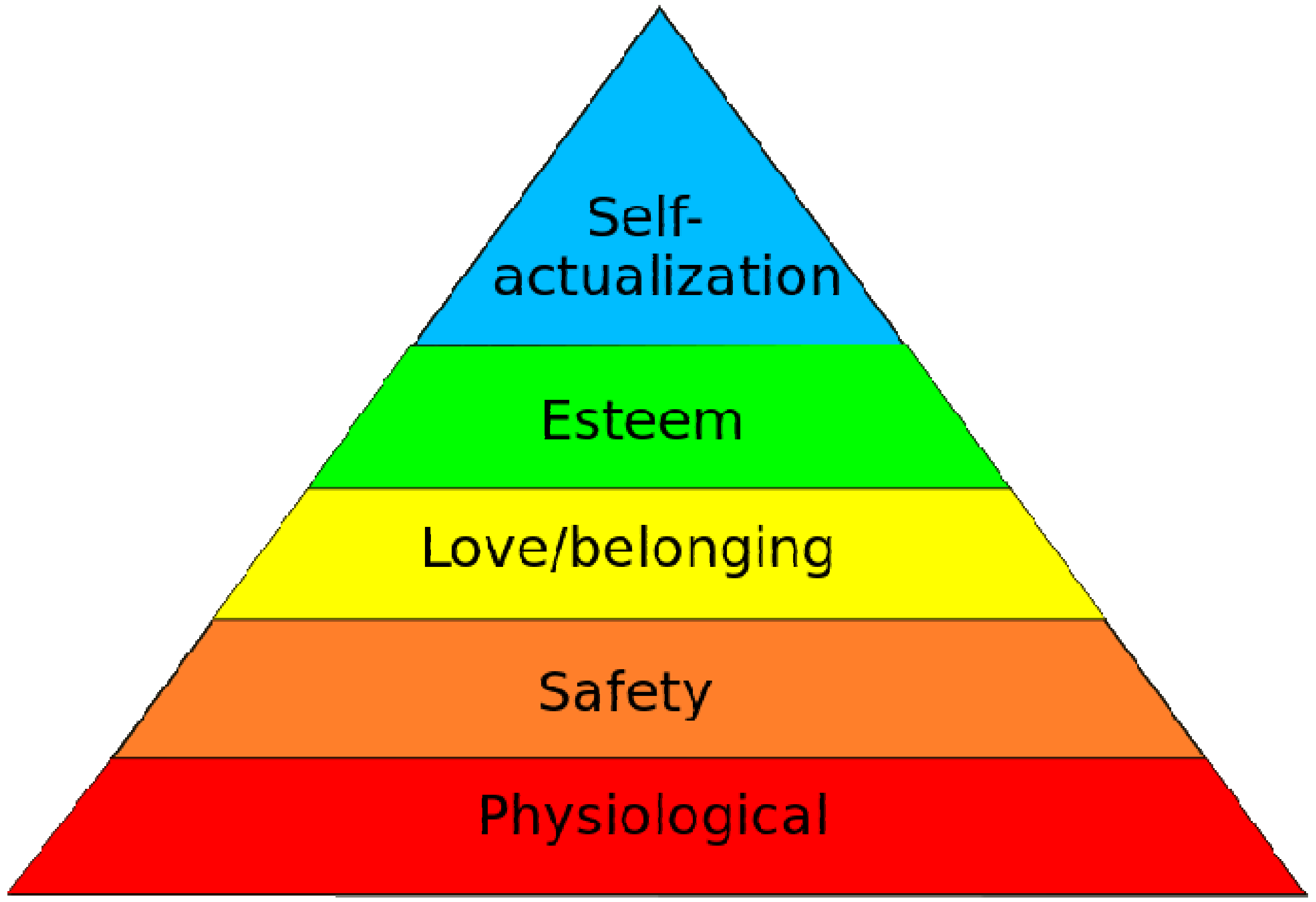
Taylor's approach: financial motivation

- Piece-rates vs. day-rates
  - In software development projects, it is difficult to isolate and quantify work done by an individual, as system development is usually a team effort.
  - Excessive distinctions between co-workers could damage moral and productivity.
  - This problem can be solved by giving bonuses to project team members at the end of successful projects.

Maslow's Approach

- Redefined motivational theory.
- Proposed a more "active" model of motivation.
- Motivations vary from individual to individual
- He argued that all people were motivated by a hierarchical set of needs.
  - Lowest level – food, shelter
  - Highest level – self-actualisation
  - As lower ones fulfilled, higher ones emerge
- Maslow's model places motivation as something within a person rather than something provided by another person

Maslow’s Hierarchy of Needs



Maslow's Hierarchy Explained

1. Biological and physiological needs - air, food, drink, shelter, ...
2. Safety needs - protection from elements, security, order, law, stability, freedom from fear.
3. Belonging needs – friendship, intimacy, sense of belonging to a group
4. Esteem needs - achievement, mastery, independence, status, dominance, prestige, self-respect, respect from others.
5. Self-Actualization needs - realizing personal potential, self-fulfilment, seeking personal growth and peak experiences.

Herzberg

Herzberg suggested two sets of factors affected job satisfaction

- *Hygiene* or *maintenance factors* – make you dissatisfied if they are not right e.g. pay, working conditions
- *Motivators* – make you feel the job is worthwhile e.g. a sense of achievement

Vroom

Vroom and colleagues identified three influences on motivation

- *Expectancy* – the belief that working harder leads to better performance
- *Instrumentality* – the belief that better performance will be rewarded
- *Perceived value* of the reward

- Motivation will be high when all three factors are high. A zero level for any one can remove motivation.

Oldham-Hackman’s job characteristics

- Identified the following characteristics of a job which make it more ‘meaningful’
  - Skill variety
  - Task identity
  - Task significance

- Two other factors contributed to satisfaction:
  - Autonomy
  - Feedback

Thank you for your attention

Any questions, please?