SS ZG622: Software Project Management (Lecture #12)

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

T V Rao, BITS-Pilani Off-campus Centre, Hyderabad

Text Books

T1: Bob Hughes, Mike Cotterel, and Rajib Mall, Software Project Management, 5th Edition, McGraw Hill, 2011

T2: Pressman, R.S. Software Engineering : A Practitioner's Approach, 7th Edition, McGraw Hill, 2010

R1: Sommerville, I., Software Engineering, Pearson Education, 9th Ed., 2010

R2: Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise by Scott Ambler and Mark Lines IBM Press © 2012

R3: Gower Handbook of People in Project Management by Dennis Lock and Lindsay Scott (eds) Gower Publishing Limited © 2013

R4: Scrum in Action: Agile Software Project Management and Development by Andrew Pham and Phuong-Van Pham Cengage Learning © 2012

R5: A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Fifth Edition by Project Management Institute Project Management Institute © 2013

R6: Jake Kouns and Daniel Minoli, Information Technology Risk Management in Enterprise Environments. John Wiley & Sons © 2010

innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

L12: Sotware Project Management –

Development of Teams, Leadership

Source Courtesy: Some of the contents of this PPT are sourced from materials provided by publishers of prescribed books

The Management Spectrum

- Effective software project management focuses on these items (in this order)
  - The people
    - Deals with the cultivation of motivated, highly skilled people
    - Consists of the stakeholders, the team leaders, and the software team
  - The product
    - Product objectives and scope should be established before a project can be planned
  - The process
    - The software process provides the framework from which a comprehensive plan for software development can be established
  - The project
    - Planning and controlling a software project is done for one primary reason…it is the only known way to manage complexity
    - In a 1998 survey, 26% of software projects failed outright, 46% experienced cost and schedule overruns

– Pressman

The People: The Software Team

- Seven project factors to consider when structuring a software development team, as per Mantei
  - The difficulty of the problem to be solved
  - The size of the resultant program(s) in source lines of code
  - The time that the team will stay together
  - The degree to which the problem can be modularized
  - The required quality and reliability of the system to be built
  - The rigidity of the delivery date
  - The degree of sociability (communication) required for the project

The People: The Software Team – Paradigms

Four organizational paradigms for software development teams

- **Closed paradigm** – *traditional hierarchy of authority*; works well when producing software similar to past efforts; members are less likely to be innovative
- **Random paradigm** – *depends on individual initiative of team members*; works well for projects requiring innovation or technological breakthrough; members may struggle when orderly performance is required
- **Open paradigm** – *hybrid of the closed and random paradigm*; works well for solving complex problems; requires collaboration, communication, and consensus among members
- **Synchronous paradigm** – *organizes team members based on the natural pieces of the problem*; members have little communication outside of their subgroups

– Constantine

The People: Challenges

- Key characteristics of modern software make projects fail
  - Scale,
  - Uncertainty,
  - Interoperability

- To better ensure success
  - Establish effective methods for coordinating the people who do the work
  - Establish methods of formal and informal communication among team members

Agile Teams

- Team members must have trust in one another.
- The distribution of skills must be appropriate to the problem.
- Mavericks may have to be excluded from the team, if team cohesiveness is to be maintained.
- Team is “self-organizing”
  - An adaptive team structure
  - Uses elements of Constantine’s random, open, and synchronous paradigms
  - Significant autonomy

Becoming a team

Five basic stages of development(as per Tuckman & Jensen):

- Forming
  - This is when the team is brought together for the first time. At this time, people tend to behave in a formal and reserved manner
- Storming
  - Team members start to position themselves against one another, often in a rather confrontational way. A leader should help build the trust among members
- Norming
  - Team members are confrontational with one another as they tackle project issues
- Performing
  - Trust is generated and team members become effective and productive working together
- Adjourning
  - Team work is completed and team is ready to disperse

Balanced teams

- Meredith Belbin studied the performance of top executives carrying out group work at the Hendon Management Centre
- Tried putting the ‘best’ people together in ‘Apollo’ teams – almost invariably did badly
- Identified the need for a balance of skills and management roles in a successful team

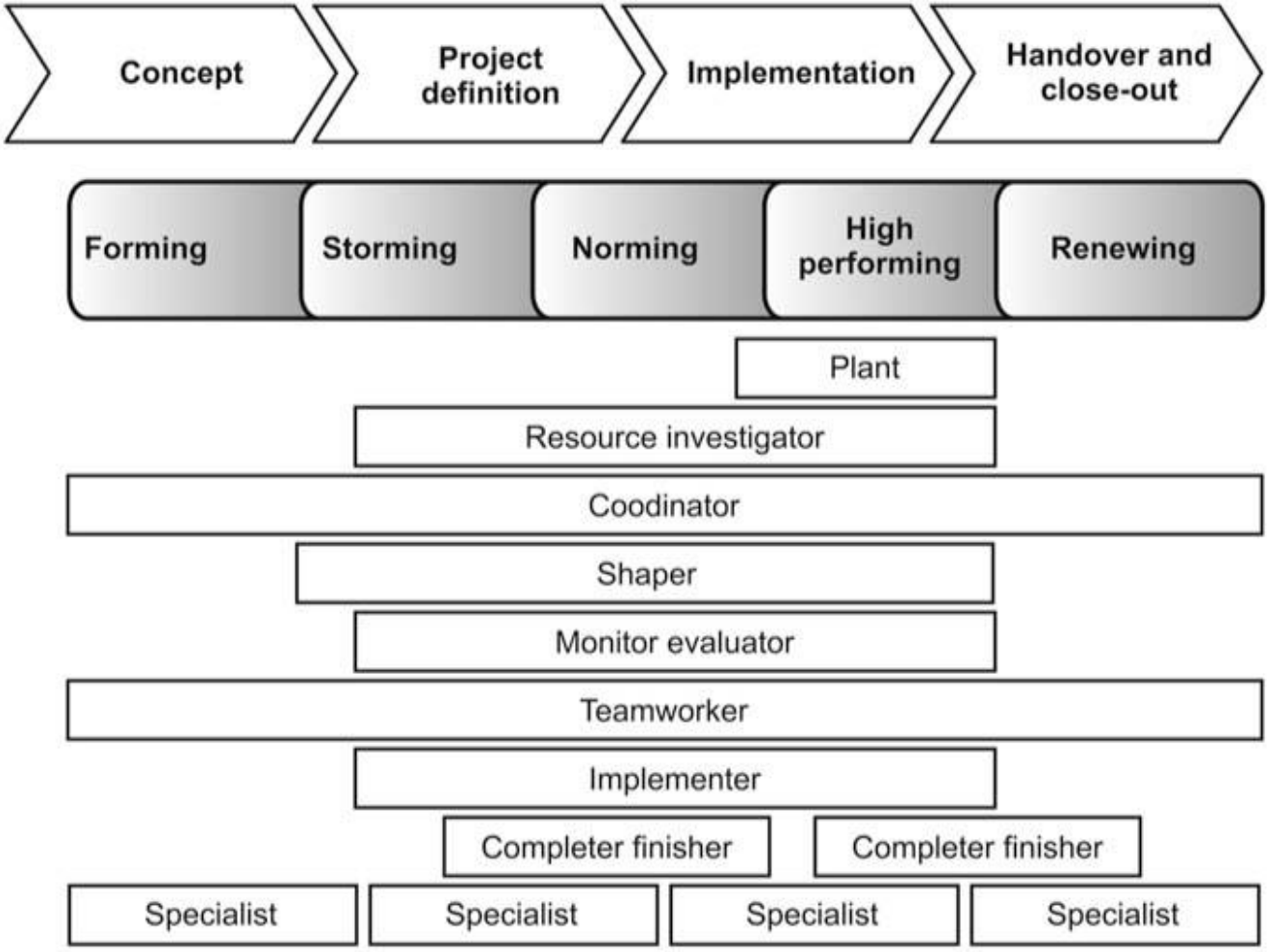
Team roles (Belbin)

- 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

Team roles (Belbin)-contd

- 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

Team roles (Belbin)



Gower Handbook of People in Project Management by Dennis Lock and Lindsay Scott (eds) Gower Publishing Limited

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

Group performance - continued

- *Disjunctive tasks* – there is only one correct answer – someone must:
  - Come up with right answer
  - Persuade the other that they are right
- *Conjunctive* – the task is only finished when all components have been completed

‘Social loafing’

- Tendency for some team participants to ‘coast’ and let others do the work
- Also tendency not to assist other team members who have problems
- Suggested counter-measures:
  - Make individual contributions identifiable
  - Consciously involve group members (‘loafer’ could in fact just be shy!)
  - Reward ‘team players’

Barriers to good team decisions

- Inter-personal conflicts – see earlier section on team formation
- 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

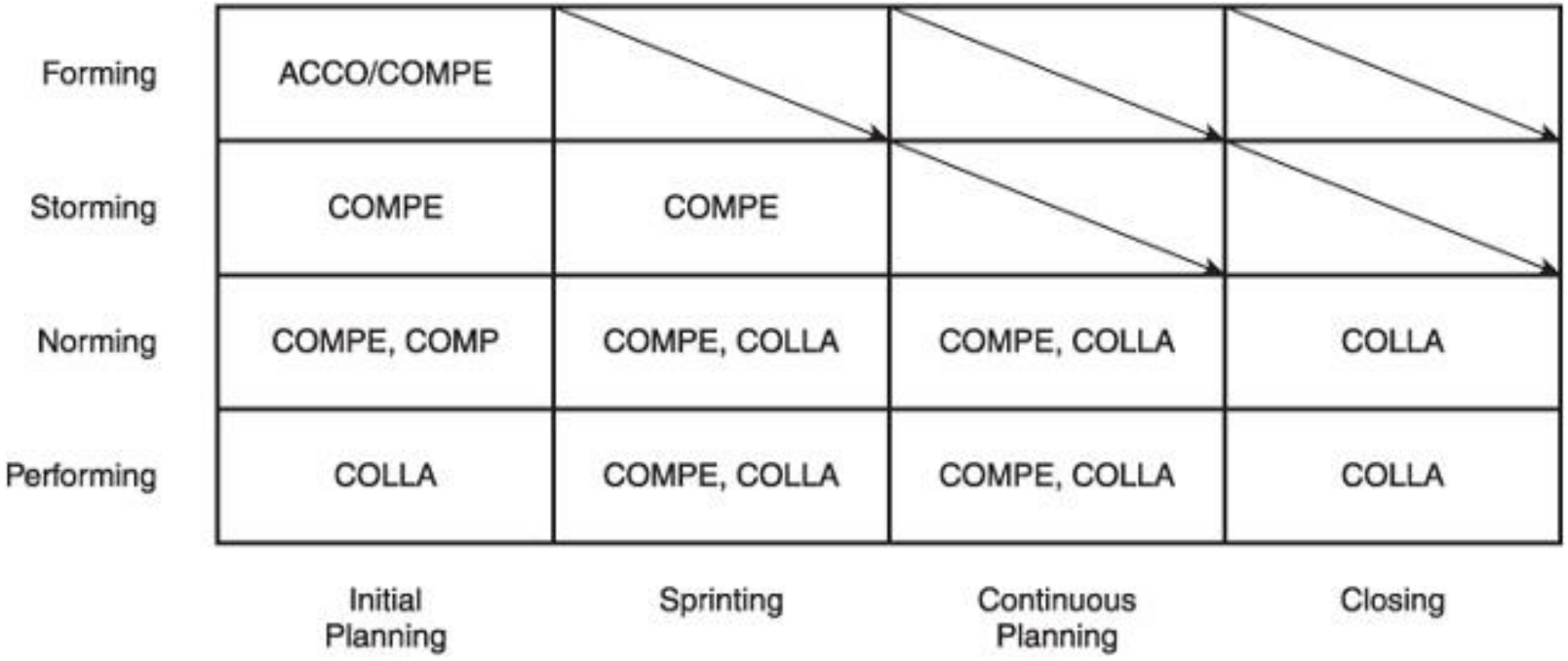
Conflict Resolution Techniques

According to Kenneth Thomas and Ralph Kilmann, common techniques are:

- Accommodating (ACCO): This technique indicates a willingness to meet the needs of others at the expense of one's own needs.
- Compromise (COMP): This happens when everyone in the conflict gives up something to reach an agreement.
- Competitive (COMPE): This is a useful technique when there is an emergency and a decision needs to be made fast or when the decision is unpopular.
- Collaborating (COLLA): All the perspectives of the different team members are examined. This technique normally leads to a good consensus.
- Avoidance (AVOID): One of the parties refuses to discuss the conflict. This is an example of a lose-lose conflict resolution technique.

Scrum in Action: Agile Software Project Management and Development
by Andrew Pham and Phuong-Van Pham

Conflict Resolution Techniques

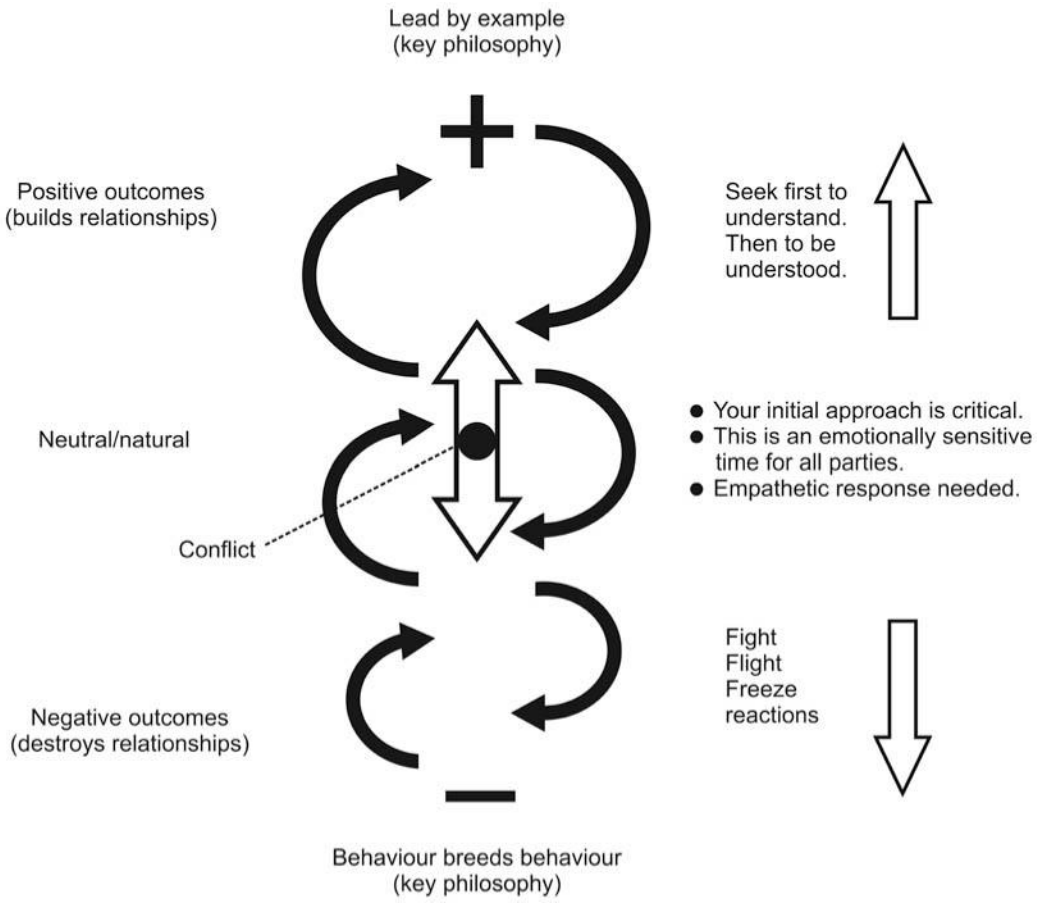


	Initial Planning	Sprinting	Continuous Planning	Closing
Forming	ACCO/COMPE			
Storming	COMPE	COMPE		
Norming	COMPE, COMP	COMPE, COLLA	COMPE, COLLA	COLLA
Performing	COLLA	COMPE, COLLA	COMPE, COLLA	COLLA

Appropriate resolution techniques to use when conflict arises in agile team

Scrum in Action: Agile Software Project Management and Development
by Andrew Pham and Phuong-Van Pham

Managing conflict with key stakeholders



Managing conflict in a way that protects project objectives implies that it also needs to be managed in a way that protects relationships with key stakeholders. Understanding of stakeholder perceptions and expressed needs and empathetic response(even if you don't agree) will put PM in a powerful position to choose an approach that is deemed best for the project.

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 moved
6. If moderator detects a consensus, stop; else back to 4

Team 'heedfulness'

- Where group members are aware of the activities of other members that contribute to overall group success
- Impression of a 'collective mind'
- Some attempts 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 key programmers, **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
- Separate documentation represents a formal mode of communication; instead XP prefers 'collective mind'
- Software regularly refactored to clarify its structure
- Test cases/expected results created *before* coding – acts as a supplementary specification
- Pair programming – a development of the co-pilot concept; also improves egoless programming

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

Scrum - continued

- Unlike XP, requirements are frozen during a sprint. At the beginning of the sprint there is a sprint planning meeting to prioritize requirements
- At end of sprint, a review meeting where work is reviewed and requirements may be changed or added to
- Scrum has elements of ‘chief programmer’ model where one owns architecture.

Co-ordination of dependencies

- The previous discussion on team heedfulness focused (mainly) in communication inside the team
- What sort of communications are needed between teams 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.

Coordination of dependencies - continued

- *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.

Why ‘virtual projects’?

The physical needs of software developers (according to an IBM report):

- 100 square feet of floor space
- 30 square feet of work surface
- Dividers at least 6 feet high to muffle noise
- Demarco and Lister found clear statistical links between noise and coding error rates
- One answer: send the developers home!

Virtual Teams - Possible 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

Virtual Teams - Further advantages

- 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

Virtual Teams - More challenges

- Possible lack of trust when there is no face-to-face contact
- Assessment of quality of delivered products needs to be rigorous
- 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

Other factors influencing communication genres

- Size and complexity of information – favours documents
- Familiarity of context e.g. terminology – where low, two-way communication favoured
- Personally sensitive – it has to be face-to-face communication here

Best method of communication depends on stage of project

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

Best method of communication depends on stage of project

- Implementation stages
  - Design is relatively clear
  - Domain and context familiar
  - Small amounts of operational data need to be exchanged
  - Favours different time/different place communications e.g. e-mail
- Face to face co-ordination meetings – the 'heartbeat' of the project

Communications plans

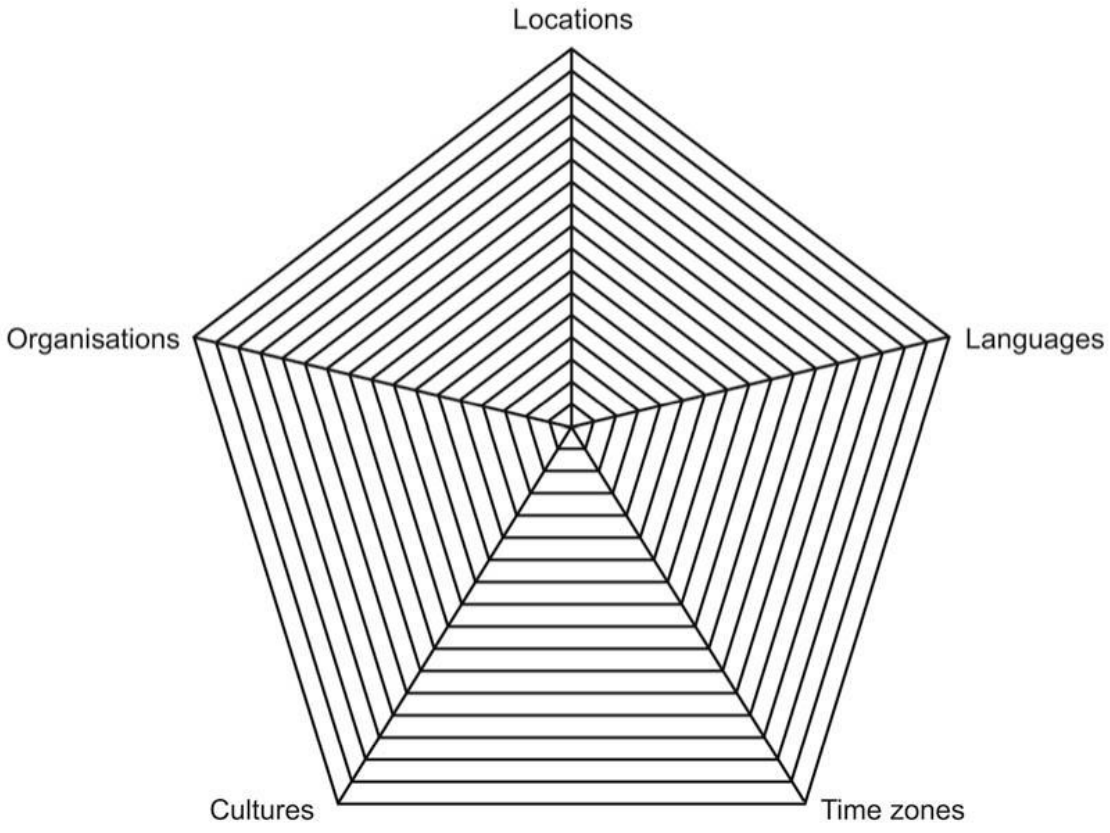
- As we have seen choosing the right communication methods is crucial in a project
- Therefore, a good idea to create a **communication plan**
- **Stages** of creating a communication plan
  - Identify all the major stakeholders for the project
  - Create a plan for the project
  - Identify stakeholder and communication needs for each stage of the project
  - Document in a communication plan

Content of a communication plan

For each communication event and channel, identify:

- *What.* This contains 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.

Dimensions of global projects



Cultures of different nations can add diversity to the work environment beyond the variety of organizational cultures, reducing the group thinking and improving the collective creativity.

Motivation can be increased when the team prefers to work in cross-cultural environments and benefit from the rich information exchange.

The diversity can sometimes be the source of conflicts and misunderstandings.

Gower Handbook of People in Project Management by Dennis Lock and Lindsay Scott (eds) Gower Publishing Limited

Global Projects

To manage projects that cross national (and cultural) boundaries, make an effort to (as per social psychologist Prof. Hofstede):

- discover the meanings of different symbols used by local people. In project management, the symbols can translate into the specialized terms, techniques and diagrams
- know their local heroes, to understand the role models of behavior
- understand and respect the rituals, e.g. the way people organize or attend meetings, local practices for celebrating success, negotiation processes and the demonstration of power when attending or rejecting meeting invitations

Global Projects

The following are examples of values that may affect global projects

- Is it polite to decline meetings because they occur during your lunch hour?
- Conversely, is it acceptable to book regular meetings during the lunch hour?
- Is it acceptable to organise a meeting starting at 6pm on a summer Friday afternoon?
- Are project managers more effective when they use their formal power (their hierarchical position) or their expert power (based on their competences)?
- What is the preferred leadership style for project managers, in each part of the project life cycle?
- How important is the performance of the team members, when compared to the way they respect and relate to their colleagues?

Cultural Dimensions as per Prof. Holstede

- **Power Distance** reflects how individuals from different cultures handle the fact that people are unequal
- **Individualism & Collectivism** the extent of group cohesiveness, the importance of participating in a social group and the values attached to working conditions and ambitions
- **Masculinity and Femininity** degree of gender differentiation
- **Uncertainty Avoidance** the resistance to change and the attitude to taking risks of individuals
- **Long-term Orientation** long-term oriented cultures tend to give importance to values such as persistence when results are slow, thrift, savings, sense of shame; short-term oriented cultures may aim to achieve quick results such as short-term targets to be accomplished before the next reporting cycle

Agile Process with Offshore Development

Martin Fowler suggests following successful practices:

- Use Continuous Integration to Avoid Integration Headaches
- Have Each Site Send Ambassadors to the Other Sites
- Use Contact Visits to build trust
- Don't Underestimate the Culture Change
- Use wikis to contain common information
- Use Test Scripts to Help Understand the Requirements
- Use Regular Builds to Get Feedback on Functionality

Article by Martin Fowler, Author and Member, Agile Alliance

Agile Process with Offshore Development (contd)

Martin Fowler suggests following successful practices:

- Use Regular Short Status Meetings
- Use Short Iterations
- Use an Iteration Planning Meeting that's Tailored for Remote Sites
- When Moving a Code Base, Bug Fixing Makes a Good Start
- Separate teams by functionality not activity
- Expect to need more documents.
- Get multiple communication modes working early

Article by Martin Fowler, Author and Member, Agile Alliance

Leadership: types of authority

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: types of 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

implementation \ decision-making	autocrat	democrat
directive	Decisions made alone, Close supervision of team	Participative decisions, Close supervision of team
permissive	Decisions made alone, Implementation freedom for team	Participative decisions, Implementation freedom for team

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)

The People: Team Leaders

- Competent practitioners often fail to make good team leaders; they just don't have the right people skills
- Qualities to look for in a team leader, as per Jerry Weinberg
  - **Motivation** – the ability to encourage technical people to produce to their best ability
  - **Organization** – the ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product
  - **Ideas or innovation** – the ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application
- Successful Team leaders use problem-solving management style
  - Concentrate on understanding the problem to be solved
  - Manage the flow of ideas
  - Let everyone on the team know, by words and actions, that quality counts and that it will not be compromised

The People: Team Leaders - Traits

- According to Edgemon, key leadership traits of an effective project manager are
  - **Problem solving** – diagnose, structure a solution, apply lessons learned, remain flexible
  - **Managerial identity** – take charge of the project, have confidence to assume control, have assurance to allow good people to do their jobs
  - **Achievement** – reward initiative, demonstrate that controlled risk taking will not be punished
  - **Influence and team building** – be able to "read" people, understand verbal and nonverbal signals, be able to react to signals, remain under control in high-stress situations

The Software Team – Toxicity Problems

- Michele Jackman observed the five factors that cause team toxicity (i.e., a toxic team environment)
  - A frenzied work atmosphere
  - High frustration that causes friction among team members
  - A fragmented or poorly coordinated software process
  - An unclear definition of roles on the software team
  - Continuous and repeated exposure to failure

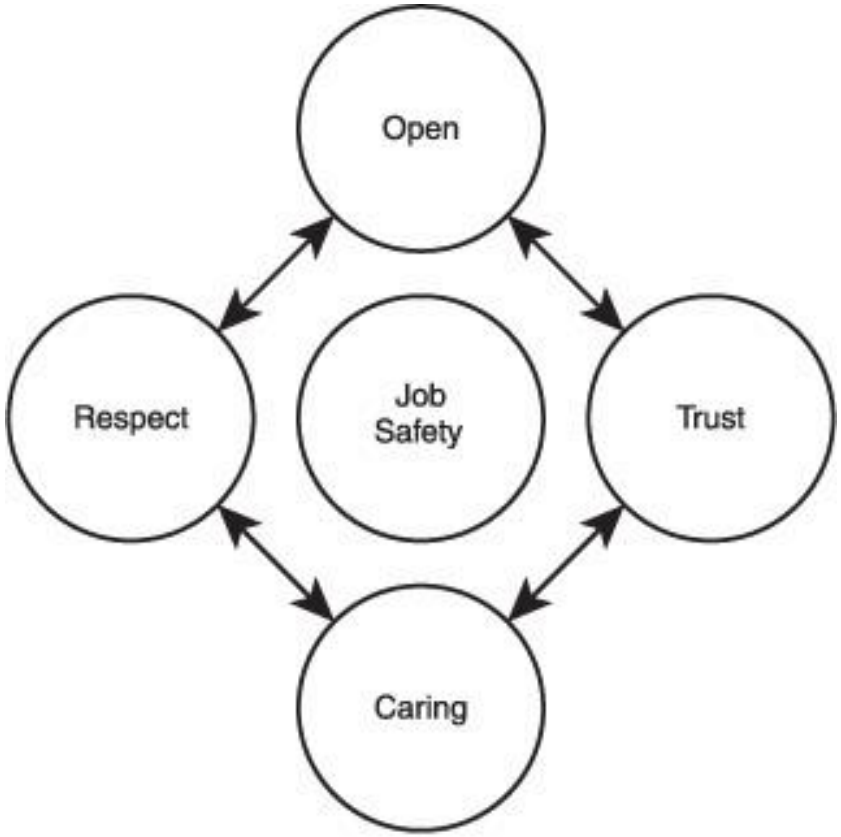
- How to avoid these problems
  - Give the team access to all information required to do the job
  - Do not modify major goals and objectives, once they are defined, unless absolutely necessary
  - Give the team as much responsibility for decision making as possible
  - Let the team recommend its own process model
  - Let the team establish its own mechanisms for accountability (i.e., reviews)
  - Establish team-based techniques for feedback and problem solving

Characteristics of performing teams

- High-performance teams are usually characterized by
  - fun, open-minded, and caring.
- Low-performance teams are usually characterized by
  - silence in meetings, forced smiles, and a cover-up attitude.
- The average-performance team
  - just go with the flow at work each day, doing only what is needed to turn in the hours and get paid.
- The only difference between the average and low-performance team is that they do not resort to cover-ups and political games.

Scrum in Action: Agile Software Project Management and Development
by Andrew Pham and Phuong-Van Pham

Conditions for great teamwork



Scrum in Action: Agile Software Project Management and Development
by Andrew Pham and Phuong-Van Pham

Thank You

Any Questions?

56