Pacific Software

Oranizational Patterns – Quick Reference

as collected by James Coplien & Neil Harrison

Project Management Patterns

Community of Trust
Human organizations are built upon a foundation of trust. This is a fundamental building block of any society, group or team. A foundation of trust is essential for fostering communication and growth.

Size The Schedule
Short schedules unnecessarily stress developers. Long schedules engender complacency. Separate the schedule for delivery to market from the development schedule, and reward success,

Get on With It (Just Do It)
If you have enough information to begin work, then don't wait around for the missing information to complete itself.

Named Stable Bases
Keeping named stable versions allows you to track progress, while maintaining stable releases. While continuous integration is important, developers need to know what works, and how.

Incremental Integration
Enable developers to build an entire release of the product against the latest source libraries with their latest changes, so that they can independently test their work before publishing their changes.

Private World (Sand Box)
Developers need their own private copy of the build environment in order for Incremental Integration to work.

Build Prototypes
When requirements & risks are not clear, build (disposable!) prototypes

Take No Small Slips
Repeated small, and unplanned slips in schedule defeat morale. When you are behind schedule, take one large slip that will accommodate for future problems as well.

Completion Headroom
Ensure that there is sufficient time between task completion deadlines and product delivery dates.

Work Split
When urgent work begins to overshadow regular development work, split the team into 2 groups, so that no more half of the team works on the urgent components, while the rest of the team continues to make progress.

Recommitment Meeting
If the minor adjustments cannot be made to realign work with the planned schedule, then reconvene the team to commit to either a new schedule, or a minimal amount of work required to finish the work episode.

Microsoft® Windows® Embedded Partner

Work Queue
When feature requirements are unclear, use short names and implied requirements in order to prioritize their order of implementation.

Informal Labor Plan
Allow developers to negotiate amongst themselves so as to create short term plans.

Development Episode
Enable isolation of the team for a reasonable episode of highly focused team-work on the current items in the work queue.

Implied Requirements
Choose and name categories of general functionality so that they may be discussed in general terms. This will allow you to progress before requirements are fully defined.

Developer Controls Process
Since the developer is a major stake-holder in all phases of development, move the developers to the hub of communication, and allow them control over the succession of activities.

Work Flows Inward
Once the developer is moved to a central communication role, it is important to ensure that the flow of communication is inwards towards the developer (the primary information client), and not outwards from the developer.

Programming Episode (Sprint)
Split work into time-boxed intervals which combine to create concrete deliverables.

Someone Always Makes Progress
When distractions prevent an even flow of work, ensure that there is always some member of the team who continues progress towards the primary goal.

Team Per Task
If a major diversion interrupts the team, create a sub-team so that only some of the team is sacrificed to handle the diversion.

Sacrifice One Person
For small diversions, sacrifice one team member so that the rest of the team can keep going forward.

Day Care
Designate one senior developer for mentoring all new incomers.

Mercenary Analyst
Developers develop. Technical Writers write. Hire an outside consultant to do documentation and technical writing.

Interrupts Unjam Blocking
If the project is (about to be) blocked because of an unavailable critical resource, interrupt that resource in order to free up the project.

Don't Interrupt an Interrupt
FIFO

Piecemeal Growth

Size the Organization
Large organizations suffer from communication break-down. Small organizations are fragile and are difficult to grow. Critical mass for a project is about 10 people.

Phasing It In
You can't always *find* the expert you need. If you can't hire them, grow them in-house.

Apprenticeship
If you have problems *retaining* expertise grow that expertise in-house from existing and new employees.

Solo Virtuoso
Small projects do not need the overhead of overstaffing. Use solo virtuosos for small projects.

Engage Customers
Incremental processes require accommodation of customer needs, as well as continuous input.

Surrogate Customer (Product Owner)
A true customer is not always available to answer enquiries, therefore it is important to fill the role of the customer with a surrogate.

Scenarios Define Problem (User Stories)
Scenarios characterize the true needs of the customer.

Firewall
Ensure that nobody pesters the developers.

Gatekeeper
Intellectual inbreeding is hazardous to your project. Have a Gatekeeper review, investigate and evaluate technologies, projects and trends from the outside world.

Self-Selecting Team
People do not work together just because they're told to. Have the team select its members.

Unity of Purpose
The team can only work together if everyone is on the same page. Common goals are critical to a team's success.

Team Pride
When a team needs to accomplish the impossible, well-grounded elitism can be productive.

Skunkworks
Projects cannot to afford to be overly innovative; however innovation is critical to an organizations future. Therefore provide time, space and resources for innovation.

Patron Role
Causes need a champion. Ensure that the project has a strategic Patron that supports the team's goals and style of work.

Diverse Groups
Diversity within a team is crucial in order to prevent "group-think".

Public Character
In healthy societies there is always someone who knows everyone (as well as what they know and do). Such a person can help smooth corners, ease work-flows and identify solution providers.

Matron Role
The social needs of a team can be met by a team member filling a motherly role.

Holistic Diversity
Systems require a diverse skill set. Since many people tend to specialize, create teams from a diverse set of specialists.

Legend Role
When a key member is set to leave the organization, train an apprentice to fill their role. The replacement can then assume a role which is named after the key person.

Wise Fool
Someone who dare speaks the unspoken can be a cynically useful tool for raising difficult issues.

Domain Expertise In Roles
Matching people to roles is difficult, at best. Create roles around the individual expertise of developers, thereby matching roles to people instead of people to roles.

Subsystem By Skill
Subsystems of a system's architecture should be defined by specific skill sets.

Moderate Truck Number
A project's Truck Number is defined by how many trucks are required to run over a developer before the project is killed. A Truck Number of 1 represents a high risk to the project. A project's Truck Number can be moderated by redundancy of skills, and distributed expertise.

Telefax: +972(4)982-4473
Rosh Hanikra,
Galil Maaravi 22825, Israel
email: info@pacificsoft.co.il

Compensate Success
Establish a range of reward mechanisms for various accomplishments. An appreciated employee is a successful employee.

Failed Project Wake
Cancelled projects should be mourned with a celebration. People invest a part of themselves in their work, and some form of closure is required when the plug is finally pulled.

Developing Pairs (Pair Programming)
Two sets of eyes are better than one. Pair programming allows coding and review to take place simultaneously. Often times, two programmers working together are more productive than two programmers working alone.

Engage Quality Assurance
For those times when developers cannot properly test their own software, engage QA as an important role in the project.

Application Design is Bounded By Test Design
In order to facilitate cooperation and interactions between developers and testers, frame the design of the application to the test design.

Group Validation
The development team, together with the Product Owner can validate design even before QA is engaged.

Organizational Style

Few Roles
If bureaucracy leads to high communication overhead, and slow response times, then reduce the number of roles in the organization (15 +/- 2).

Producer Roles
In order to facilitate the identification of Roles which should be removed, divide the organization's roles into 3 categories: Producers, Supporters and Deadbeats. Eliminate the Deadbeats. Further reduction can be achieved by combining Supporter roles.

Producers in the Middle
Producer roles should be the focal point of communication.

Stable Roles
Handling of a disruption to a project should be treated as a temporary role. Individuals who fill these temporary roles should maintain their primary roles within the organization.

Divide & Conquer
When the organization outgrows its ability to effectively communicate, then divide it into smaller organizations.

Conway's Law
Product Architecture follows Organizational structure, follows Business Domains.

Organization Follows Market
If the business lacks accountability to a given market, then designate specific organizations to specific markets.

Face To Face before Working Remotely
Geographically divided projects benefit greatly when they begin with project members all meeting together.

Form Follows Function
Specific domain expertise or even familiarity with given software artifacts and processes should be assigned a role. This prevents confusion and enables single point of contact for related issues.

Shaping Circulation Realms
Roles and environments should reflect required communication paths.

Distribute Work Evenly
Distribute workload evenly throughout the organization. An uneven workload causes burnout and typically signifies more severe organizational problems.

Responsibilities Engage
If central roles are overloaded it is possible to reduce the pressure on them by increasing the communication paths between adjacent non-central roles.

Hallway Chatter
If a team is separated from other teams, informal communication is drastically reduced, thereby adversely affecting productivity. Placing outer roles in close proximity to central roles creates "hallway chatter" and a better working environment.

Decouple Stages
Overlapping stages of development often causes rework. Decouple and serialize stages with distinct hand-offs between each stage.

Hub, Spoke and Rim
In a high-context environment, decoupling of stages may require a central hub-role to manage the handoffs between the various development stages.

Move Responsibilities
Moving responsibilities from one role to another helps separate tightly coupled roles.

Upside-Down Matrix Management
If skills and resources are not properly matched with a particular aspect of the work, then break through organizational structure to leverage external individuals, teams and resources in order to produce the proper results. Create new groups using the right people irregardless of where they come from within the organization.

The Watercooler
Make space for simple human interactions in order to foster informal communication paths.

Telefax: +972(4)982-4473
Rosh Hanikra,
Galil Maaravi 22825, Israel
email: info@pacificsoft.co.il

Three to Seven helpers per Role
Over an extended period of time, people are capable of maintaining 7 +/- 2 effective communication relationships. If a role tries to maintain more contacts on a regular basis, work and communication will suffer.

Coupling Decreases Latency
Tightly coupled roles have low latency between them. In order to produce fast results between 2 roles, place them adjacent to one another.

People and Code

Architect Controls Product
For long term projects, and architect is important in order to maintain architectural style of the product.

Architecture Team
If the product is too large and complex for a single individual to thoroughly understand its components, use an architecture team.

Lock 'em Up Together
If a team cannot reach a breakthrough in architecture or design, isolate them for several days so that they may work uninterrupted.

Smoke-Filled Room
It is sometimes necessary to make a decision quickly, without sharing the deliberation process with certain individuals. A public decision can be made in private, without sharing its rationale.

Stand-Up Meeting
Short stand-up meetings keep people informed, and are an effective method for floating information to the surface.

Deploy Along the Grain
Allow people to have long-term responsibility for specific parts of the system. This enables them to optimize and reuse modules efficiently.

Architect Also Implements
Architects need to be involved with the day-to-day implementation, otherwise they will lose touch with the reality of product development.

Generics & Specifics
In a rapidly growing organization, assign generic functionality to senior and more experienced employees, and have the newcomers to the specific assignments.

Standards Linking Locations
Establish standards (interfaces, etc) to allow geographically separated teams to work together.

Code Ownership
Code should only be modified by its owner. If nobody owns a piece of code, assign ownership to someone.

Feature Assignment
The easiest way to partition a large project is by assigning features to individuals.

Variation Behind Interface
Create stable interfaces to encapsulate variability.

Private Versioning
Providing individual developers with their own private revision control, allows them to avoid publishing intermediate results.

Loose Interfaces
Loose interfaces, such as callbacks, allow for loose coupling between modules, thereby insulating them from each other's changes.

Subclass Per Team
If a module causes a clash between two subsystems, then subclass the module so that each subsystem has its own layer in the hierarchy.

Parser Builder
Factory Pattern for input streams.

Arranging the Furniture
Assigning newcomers the task of rearranging existing code is an effective way for them to stake out their intellectual territory, and have them assume code ownership.

Telefax: +972(4)982-4473
Rosh Hanikra,
Galil Maaravi 22825, Israel
email: info@pacificsoft.co.il