The XP Project as a Complex Adaptive System

Bob Payne and Sanjiv Augustine

Projects adjust in response to external and internal changes and seem to have a life of their own. Although this is brushed aside as cliché, we intuitively know that the interactions of project participants and other factors manifest the dynamic nature of projects. This dynamic behavior of a project that is greater than the sum of its constituent parts drives us to look at projects as Complex Adaptive Systems (CAS). CAS self-organize and adapt to changes in the environment without central rules governing their behaviors. This robust adaptive behavior is an emergent property of interactions among the sub-parts and/or between the environment and the system.

Agents - Agents are the semi-autonomous building blocks in a CAS: they seek to maximize some measure of goodness, or fitness, by evolving over time. Simple, local rules guide the interaction between agents of a system and result in global, complex behavior. Several factors affect how people work together in an XP Project to create innovative, emergent outcomes: *internal models* direct behavior; *diversity* drives innovation and self-organization; *tagging* enables easy organization; *strategy* dictates influences team interaction; and *building blocks* provide abstractions necessary to parse and organize the environment.

Complex Adaptive Systems (CAS)

Complexity science postulates that systems can be understood by looking for patterns within their complexity, and this understanding used to describe potential evolutions of the system.

CAS self-organize and adapt to changes in the environment without central rules governing their behaviors. This robust adaptive behavior is an emergent property of interactions among the sub-parts and/or between the environment and the system.

Agents are the semi-autonomous building blocks in a CAS: they seek to maximize some measure of goodness, or fitness, by evolving over time. Simple, local rules guide the interaction between agents of a system and result in global, complex behavior.

Internal Models. CAS principles state that internal models are important in determining how people react to a set of circumstances. The XP Values – c*ommunication,* s*implicity, feedback, and courage* – serve as touchstones for making day-to-day decisions and aligning internal models:

1. Communication – Shared information creates a workplace where self-organized, highly interactive, emergent, dynamic decision-making can occur;
2. Simplicity – Complex solutions and rules lead to misunderstanding, over engineering, project inflexibility, and low compliance within the group. XP stresses simplicity in all aspects of the project to facilitate clear communication and elegant design;
3. Feedback – Direct low-latency feedback is crucial when significant changes in the project are needed without incurring inordinate loss of time/money; and
4. Courage – Enables rapid self-organization, full participation from all team members, rapid overarching changes to the project or code base, and is necessary to allow teams to stay focused during times of stress.

Diversity. CAS rely on the diversity of skills, strategies and experience of agents to create dynamic, adaptive problem solving behavior. Biological models illustrate that greater diversity among the agents can improve the chances of success or survival. XP relies on the diversity of experience and skills among team members to facilitate self-organized, dynamic problem solving and emergent leadership.

Tagging. Agents form close interactions with agents they perceive to be like themselves. In CAS, the term *tags* or *tagging* is used to describe the agent attributes used for agent identification within the system. Not only do individuals represent CAS agents, but agents can group to form *meta-agents*. The creation and interaction of these composite meta-agents is directly driven by the tagging symbols used to create them on a project. XP breaks the team into a limited number of principal roles (*Developer* and *Customer*) and supporting roles (*Manager*, *Coach* and *Tracker*). The Developer and Customer tags are reinforced by a *Bill Of Rights* to create two meta-agents within the team with different practices and strategies for interaction.

Strategy. Planning is the XP practice most profoundly affected by game theory and strategy. The implementation of two simple rules turns long-term (*Release*) and short-term (*Iteration*) planning activities into constructive negotiation: Developers solely own task estimation, and Customers solely own task creation and prioritization. Developers try to maximize the number of bug-free features while Customers try to maximize the value of the features in each iteration. Thus, Customers and Developers cooperate strategically for economic maximization, rather than jockey for power.

Building Blocks. Building blocks are abstractions that agents use to organize the world by grouping things and reassembling them in ways that suit a given problem. CAS agents create differing ecosystems of building blocks for each instance of a model. Similarly, it's normal for the practices and values of XP to be put together uniquely for each project; the solutions change but the building block abstractions remain primarily stable.

Agent Interactions - The power of CAS is revealed in agent interactions. It isn't driven by command and control, but emerges from the interactions of many individual, distributed agents making decisions based on internal models, available information, and a simple set of behavioral rules. This is similar to the complex behavior of ant colonies. Characteristics of agent interactions in CAS are: *Simple Rules, Aggregation and Emergence, Feedback and Non-linearity.*

Simple Rules. Simple rules in XP have a dual effect. They provide enough control to keep the system from spiraling out of control and enough freedom to construct the product with minimal interference; thus utilizing the CAS nature of all projects rather than suppressing it.

CAS Principle	XP Manifestation
Agents maintain *internal models* that direct their behavior.	XP values serve as an internal model.
Skill *diversity* among agents contributes to innovation and self-organization.	Collaborative practices (collective ownership, etc) enable diverse skills/experience.
Tagging enables easy identification and organization.	Limited primary roles of *customer* and *developer* allow easy tagging.
Strategy dictates cooperation over competition.	Game theory provides optimization in planning activities.
Building Blocks provide necessary abstractions to organize the environment.	XP values and practices provide a simple set of concepts to tackle most projects.
Local, strategic rules support *aggregation* and *emergence* in a team environment.	XP values and practices form the basis for complex behavior.
Emergent order is a bottom-up manifestation of order, while imposed order is a top-down manifestation.	Simple rules, disciplined coding practices and reduced hierarchy lead to *self-organization, emergent architecture/design,* and *stability in the face of change.*
Feedback enables change and adaptation.	Constant feedback through *tracking, frequent releases co-location, paired programming,* and the *daily standup* enable change and adaptation.
Non-linear dynamical systems are continuously adapting when they reach a state of *dynamic equilibrium* termed the *edge of chaos.*	Sweeping changes can be rapidly accomplished utilizing *unit tests, refactoring, co-location* and *continuous integration.*

Aggregation and Emergence. These are phenomena in CAS where the interactions of agents create something larger than the sum of their contributions. In XP, they're manifested through self-organization, emergent architecture and design; and stability in the face of change:

1. Self-Organization – XP provides almost no hierarchy within the team but instead provides counterbalancing roles with limited power. The team organizes around the natural leaders that emerge to solve a given problem. XP supports self-organization through: *Co-Location*, *Paired Programming, Collective Ownership and Accepted Responsibility.*

2. Emergent Architecture and Design – XP relies on and encourages the process of emergent architecture and design through the following XP practices: *Test Driven Development, Refactoring, Simple Design, Pair Programming,* and *Co-Location.* Requirements and the code on an XP project co-evolve, moving the code base to a state that appears perfectly "designed" for current requirements.

3. Stability in the face of Change – XP strives to be more adaptive and reflective of reality rather than deterministic by allowing scope changes at iteration boundaries. It relies on short iterations, and simple quantitative measurements as inputs to the planning process. This information is invaluable in planning and allows the project to remain stable.

Feedback. CAS agents use positive and negative feedback from the environment and other agents to measure their effect on the system, as input for the development of their internal models, and for development of strategies for future problems. Unit tests and continuous integration are examples of negative feedback – they slow down sloppy developers. *Co-location*, *daily standup* and *pair programming* can result in positive feedback that pushes the system toward change. Frequent, objective *tracking* and releases allow all participants to know the project status at any given time.

Non-Linearity. XP projects can absorb radical shifts in scope and requirements that would force other projects into a state of chaos. XP supports these sweeping changes through development practices emphasize simplicity and the ability to refactor the code (*test driven development, refactoring, continuous integration)* and by allowing the team to deploy tackle the problem with all the skill available on the team (*co-location, whole team).* XP also has a lower reliance on documentation reducing the overhead of document change management. CAS Agent interaction can also create non-linear artifacts amplifying actions through increased interaction and feedback. XP project rely on *co-location* and *feedback* to increase team member interaction, this increase in interactions can create disproportional effects within the team. Therefore it is critical that XP teams be made up of co-operative rather than disruptive individuals.

Bob Payne is CEO and Founder of Electroglide Inc., a Washington DC firm specializing in XP project implementation, development and training. He holds a MSEE in computer architecture for artificial intelligence. Bob is a member of the ACM, the Agile Alliance and co-founder of the Washington DC XP users group. Contact him at bob@electroglide.biz.

Sanjiv Augustine is Director, Technology at CC Pace Systems, Fairfax, Virginia. He is a leading agile management practitioner, consultant and speaker. Over the past few years, he has deployed XP at large financial institutions, and has managed several XP projects varying in size from five to over one hundred people. Contact him at sanjiv.augustine@ccpace.com.