



Robert Watkins

Robert Watkins is an Engagement Lead for Changeis and Leader of the Enterprise Technology Solutions Community of Practice. He has more than twenty years of experience working in large corporations in a range of software engineering roles. Along the way, he has taught both graduate and undergraduate courses in Mathematics and Computer Science. In his free time, he likes camping in the woods, reading and learning new things: from the Mathematics of Blockchain to 3D printing, to welding and learning Danish.

Contact Changeis

www.changeis.com

info@changeis.com

(703) 348-9669

What's Your Problem?

"If your only tool is a hammer, every problem looks like a nail."

– Unknown

This is the first in a series of articles to bring you tools you can use to improve your problem-solving skills.

Solving problems is something we do so frequently, we rarely think about the process we use.

We often bring in consultants or contract staff because there is a specific problem to solve. Even if the challenge is well known, the obstacles to overcome may not be apparent. As these obstacles unfold, strategies for dealing with them must be quickly established. As we all know, not all problems are created equal. So how do we get pointed in the right direction?

The Model

I start with the **Cynefin Model of Problem Types**. The Cynefin Model is a framework based on complexity science. It allows for problems to be assigned to categories defined by the nature of the relationship between cause and effect. Four of these categories—simple, complicated, complex, and chaotic—can help leaders diagnose situations and act in contextually appropriate ways.

Complexity science is a way of looking at systems for characteristics such as:

- Interacting elements which are linear or nonlinear; minor changes in nonlinear systems can produce disproportionately major consequences.
- The dynamics of a system. If the whole is greater than the sum of its parts, then solutions cannot be imposed; rather, they arise from the circumstances.

Another way to consider the categories is to look at the constraints of the situation to narrow down what options to consider.

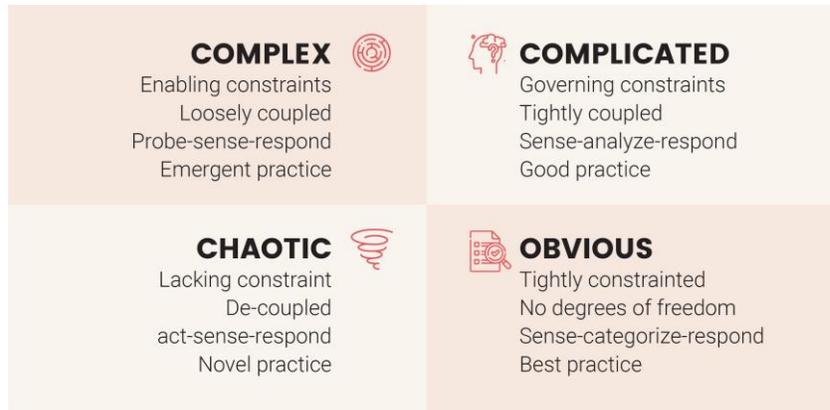
Obvious – Problems that are tightly constrained

Complicated – Problems related to governing constraints

Complex – Problems related to enabling constraints

Chaotic – Problems lacking constraints

Cynefin Model of Problem Types



An Example

One of the tasks that Changeis has addressed is **standardization of the build and deploy process for IT applications**. Depending upon the application, there are varying paths to a solution. Let's look at a few that we've worked through:

Cloud-native Application

We were asked to build an Application Programming Interface (API)-based data layer to enable better access control and allow the backend data source to ultimately be migrated to a different database platform. The agency already had Amazon Web Services and Continuous Integration / Continuous Delivery (CI/CD) capability.

In this case, the scenario is tightly constrained by well-defined, existing processes. It is a known amount of work with a clear recommendation for a process. This is an **Obvious** context and best practices are the appropriate way to move forward.

Legacy Application with Automated Deployments

One web application we had the responsibility of managing was built using an older version of .NET. There was an existing CI/CD job that would build the application, run unit tests and then deploy it.

This was a **Complicated** problem as the build and deploy process was generally well understood but coordinating any changes to the CI/CD jobs proved to be tedious. As opposed to the cloud-native applications that were built for automation, legacy applications must contend with a variety of server idiosyncrasies when automating the deployments.

Modern Server Application Coupled with COTS Solution

Another system we encountered was an application that was built (mostly) with Java, but also included a Commercial Off the Shelf (COTS) product that provided a portion of the functionality. This COTS product was initially selected because it allowed non-technical users to quickly make changes without creating a new software release.

Automating the build and deployment of the Java components remained straightforward. Although the COTS solution was originally selected because it allowed non-technical users to modify it, those users eventually relied on the development teams to make those changes. This meant discovering a way to extract the manual changes programmatically from the system, packaging that up to be placed in the code repository and programmatically applying those changes to other application environments. This involved trial-and-error and testing to ensure functionality (The Cynefin Model refers to this process as 'probe-sense-respond'). This was a **Complex** problem.

Legacy Application with Manual Deployments

Prior to the common practice of CI/CD or even Application-Release Automation, application releases involved manually performing the various steps that are now automated. We were given the responsibility for updating such an application. The application made heavy use of PL/SQL packages, batch jobs run under cron, image files to support report headers, binary-format report definition files and often, Apache configurations. This application used a platform that was not designed for automated deployments.

This is a **Chaotic** problem and is a work in progress. For example, when we originally automated the file copy for the report images, we discovered that there were not only one but two additional report template frameworks being used that assumed all the images were in a different location. Later, we automated the Apache configuration changes only to find that the development environment only had a single server, while production had multiple. There are other issues as well. This work is ongoing, yet a rewarding challenge for our expertise.

The Cynefin Model can be applied to all sorts of problems, not just these scenarios. You can be prepared for solving problems by knowing more about what kind of problem type you have ahead of time.

If your problem looks like this...	It's likely this kind...	Use this approach...
Highly-constrained or simple, like our 'Cloud-native Application example	Obvious	Use Best Practices
Lots of moving pieces that require coordination, like the Legacy Application example	Complicated	Use Good Practices
The work is generally known, but has no precedence, like the Modern server with COTS example	Complex	Use an Emergent Practice. This is where you use "probe-sense-respond"
Where there is no precedence and the work does not have a shared understanding, like the Legacy Application with manual deployments	Chaotic	Develop a completely new, or Novel Practice, which will require continuous tweaking

The more you know about the type of problem you are trying to solve, the better you are armed to choose a vendor that is well-suited to solve that kind of problem.

REFERENCES

- The Cynefin Center's deeper dive into this Framework - <https://cynefin.io/wiki/Cynefin>
- Farnam Street's blog entry on Mental Models - <https://fs.blog/mental-models/>
- Harvard Business Review article on the Cynefin Framework - <https://hbr.org/2007/11/a-leaders-framework-for-decision-making>

Changeis' Core Capabilities

Strategy & Change Management

Changeis develops and executes strategies to maximize our customers' success. We apply in-depth industry knowledge, analytics expertise, and strategic acumen to design the right solution for their unique needs. Once the strategy is in place, we help customers communicate these changes and promote adoption among stakeholders.

Large Scale IT Solutions

In the age of digital business, organizations need highly flexible and responsive IT capabilities. This often means modernizing their legacy technology.

Changeis realizes large-scale IT transformation projects from conception to implementation – resulting in efficiency, automation and streamlined workflows through digital operations.

Investment & Acquisition Management

We help organizations obtain the resources they need to implement their strategic plans and initiatives. Rigorous data management, analytics tools, and compelling writing form the bedrock of our deliverables for customers.

Program Management Office (PMO) Support

We provide governance to ensure that strategies are implemented effectively, and that deliverables support organizational goals. Our services include resource management, risk management, performance management, and portfolio and program development and management.

Innovation & Optimization Management

There are major optimization opportunities within every organization. We help our customers identify and capitalize on them. Examples of some of our toolkit include Life Cycle Planning and Support, Technology Strategy and Systems Integration, Business Intelligence and Supply Chain Management.