DevOps Nirvana
What Is It, and How Do I Get There?
DA DAY 2018 – May 15th

- A full day online conference focused on advances in DA and client success
- Expert Presenters
- New DA techniques and learning materials
- Receive DA certification renewal credits for a full year
- Cost: $50 for Active Members, $100 for Non Members

- Call for Speakers now open – contact Conference@DisciplinedAgileConsortium.org

Rolf W. Reitzig

Rolf W. Reitzig is a member of the IT Consulting team at Sirius, a group of highly experienced ex-CIOs/CTOs/VPs, business owners, and management consultants that work strategically with Sirius clients to design custom, technology-agnostic solutions addressing critical business needs.

For more information, contact Rolf at Rolf.Reitzig@Siriuscom.com
The Disciplined Agile Framework

http://www.disciplinedagiledelivery.com/the-agile-tractor-engine-analogy/

© Disciplined Agile Consortium
Business Drivers

Software and IT organizations are under pressure to develop & deploy more software, yet many companies are losing precious time-to-market due to manual processes and disparate, non-integrated development tools & IT Infrastructure.
What is DevOps?

- DevOps is a framework and organizational culture that focuses on reducing software time to market by accelerating flow and seamless integration between Software Development and Operations teams.
- DevOps evolved because businesses were demanding more from Development teams but Operations were having difficulty accommodating the faster rate of software development.
- DevOps impacts every layer of the dev/test/prod technology stack as well as the software development lifecycle - it is not a single software, hardware, or service solution.
- **DevOps reduces software time to market, defects, and risk.**
What is DevOps? In a nutshell, a System for Changing Systems to get new features to users faster!

- **Accelerate** the time to value of your new business ideas
- **React** to market opportunities more quickly
- **Beat** competitors to market with new capabilities
- **Remove** friction from your software delivery cycles
- **Eliminate** organizational silos that slow business down

Adopting DevOps practices and tools reduces build/deploy/release cycles from days/weeks to minutes/hours and frees up resources to work on new features and functionality!
Value of DevOps Nirvana – High vs. Low Performers

- **200x more frequent deployments**
- **24x faster recovery from failures**
- **3x lower change failure rate**
- **2,555x shorter lead times**

<table>
<thead>
<tr>
<th>High Performers</th>
<th>Low Performers</th>
</tr>
</thead>
<tbody>
<tr>
<td>New work: 49%</td>
<td>New work: 38%</td>
</tr>
<tr>
<td>Unplanned work or rework: 21%</td>
<td>Unplanned work or rework: 27%</td>
</tr>
<tr>
<td>Other work: 30%</td>
<td>Other work: 35%</td>
</tr>
</tbody>
</table>

Employees in high-performing teams were 2.2 times more likely to recommend their organization as a great place to work.

Data from the 2016 State of DevOps Report
DevOps Nirvana

1. Technical and organizational architectures are loosely coupled, breaking down interdependencies to allow for faster value creation
2. Small teams of developers independently implement atomic features in production-like pre-prod environments available via self-service in real-time
3. Code changes are small, limiting work in progress and shortening lead times
4. When code changes are committed into version control, software is auto-built and deployed to production-like pre-prod environments
5. Once deployed, batteries of code/design quality scans & automated tests are run, giving continual assurance that the code and environments operate as designed and are always in a known, secure, and deployable state, while also allowing almost immediate identification and correction of defects
6. With little to no delay once complete and auto-tested, code is auto-deployed into production environments quickly, consistently, safely, and securely
7. Code deployment is completely decoupled from code release
8. When problems are found, teams swarm to correct them immediately and then implement appropriate automated tests to identify them earlier going forward
9. Pervasive production telemetry in both code and environments ensures user experiences are optimized, and problems are detected, reported, and even auto-corrected quickly
10. Via fast and continuous feedback loops, development teams are always gaining insights into and improving their development processes, environments, the software product, and the customer experience

Adapted from The DevOps Handbook, 2016
Key DevOps Nirvana Concepts

1. Make Work Visible
2. Limit Work in Process
3. Reduce Batch Sizes
4. Reduce Handoffs
5. Continually Identify and Elevate Constraints
6. Eliminate Hardship and Waste
7. See Problems as They Occur
8. Swarm and Solve Problems to Build New Knowledge
9. Push Quality Closer to the Source
10. Enable Optimizing for Downstream Work Centers
11. Enable Organizational Learning and a Safety Culture
12. Institutionalize the Improvement of Daily Work
13. Transform Local Discoveries Into Global Improvements
14. Inject Resilience Patterns Into Daily Work
15. Leaders Reinforce a Learning Culture

From The DevOps Handbook, 2016
Disciplined DevOps

http://www.disciplinedagiledelivery.com/disciplineddevops/
Critical Components of DevOps Nirvana - Dev

**Agile Requirements Management** - Requirements are managed in an Agile fashion with Feature and/or User Story forced-ranked backlogs, avoiding Big Requirements Up Front (BRUF), customers and stakeholders are continually involved during the development effort, providing as-needed validation and course correction as the system evolves.

**Agile Work Management** - Software development is managed in an Agile or Kanban fashion with small independent teams focusing on short delivery cycles.

**Development, Change, and Configuration Management** - Software is developed in production-like development environments built on-demand, with all components related to the system (OS, patches, middleware, data, tools/compilers, settings, application software, etc.) configuration managed, modifications to the entire system tracked with change requests, and binaries stored and managed in a binary repository.

**Continuous Integration and Delivery** - When changes are delivered, the system is immediately automatically built and deployed to the appropriate production-like pre-prod environments built on-demand.

**Automated Testing** - Batteries of automated code/design quality scans & unit, integration, functional, security, performance, etc. tests are immediately run upon deployment to pre-prod environments.

**Agile Architecture and Design** - The system is architected in a loosely coupled fashion allowing for changes to be delivered atomically versus monolithically.

**Agile Portfolio Management** - The entire organization is architected and managed in an Agile fashion, with ongoing value streams instead of fixed scope/duration/cost projects, and with empowered teams of generalizing specialists focused on delivering small units of value continuously.

**Automated Organizational Dashboards** - Instantaneous insight is provided into all aspects of the software development value
Critical Components of DevOps Nirvana - Ops

Cloud Management - Infrastructure is virtualized via private, public, or hybrid cloud hypervisors and/or container management systems

Infrastructure as Code (IaC) and CMDB - All aspects (os, patches, middleware, DB, application software, configurations, etc.) of all environments are configuration managed using IaC including servers, routers, switches, etc. Deployment of changes are managed and deployed automatically, and unapproved changes to environments are automatically detected and rolled back or alerts raised.

Production Support - An ITSM system is used to track and manage incidents, problems, defects, etc. to resolution

Operations Monitoring - All aspects of the system (os, middleware, DB, application software, user experience, etc.) are monitored and tracked, and alerts are raised when user experiences are degrading or issues are detected

Operations Automation - Simple system corrections or other operations are automatically employed to address issues raised without human intervention

Operations Orchestration - Atomic automation capabilities are combined to automatically address more sophisticated system corrections or operations without human interaction

Employee Self-Service - Employees can request and have automatically fulfilled many IT requests such as environment provisioning, software downloads, etc.

Automated Organizational Dashboards - Instantaneous insight is provided into all aspects of system operations
Sirius DevOps Maturity Model

1. Adhoc development, tools, processes, training, inconsistent development results across projects and releases. No common tools, processes, or activities can be expected.

2. Agile work management, requirements, change, and configuration management in place, automated continuous integration and deployment.


4. Agile Portfolio Management in place.

5. Integrated & automated data, metrics, tools, processes across all Development activities. Integrated & automated data, metrics, tools, processes across all Operations activities.

Development

Operations
Sirius DevOps Reference Architecture

Concept to Cash Value Flow

Business Process

Customer Needs
Concept Def
Solution Dev
Test
Release & Deploy
Provision
Monitor
Event/Prob Mgmt
Customer Feedback

Governance Frameworks, Practices, & Tools

Enterprise Agile (DA, SAFe), CMMI
Agile Portfolio Management
Agile Work Management
Agile Reqs Mgmt
Agile Arch & Design
Dev, Change & Config Mgmt
Cont Int & Delivery
Automated Testing
Virtualization Mgmt
IaC & CMDB
Automation

Security, Data, Infrastructure, People, & Teams

Largely In Place
Somewhat In Place
Largely Not In Place

KEY

DISCIPLINED AGILE CONSORTIUM
Mapping to Sirius DevOps Reference Architecture

DEV | Concept to Cash Value Flow | OPS

Customer Needs | Concept Def | Solution Dev | Test | Release & Deploy | Provision | Monitor | Event/Prob Mgmt | Customer Feedback

Enterprise Agile (SAFe, DAD), CMMI

Portfolio Management

Agile Work Management

Agile Reqs Mgmt | Agile Arch & Design | Dev, Change & Config Mgmt | Cont Int & Delivery | Automated Testing

Cont Int & Delivery

Automated Testing

Cloud Mgmt

IaC & CMDB

Monitoring

Orchestration

Self Service

KEY

Largely In Place

Somewhat In Place

Largely Not In Place

DISCIPLINED AGILE CONSORTIUM
Holistic & Encompassing System Configuration Management
Environment Configuration & Pipeline Management
DevOps Nirvana Key Tooling Value Flow Diagram

1. User/PM/PO
2. Developer
3. SCMM/Build
4. IDE & Env
5. Binary Repository
6. Test
7. Test Automation (unit, functional, performance, security, etc.)
8. Test Mgmt
9. Stage
10. Prod
11. Virtualization
12. ITSM
13. ALM System

DISCIPLINED
CONSORTIUM
Questions
Thank You!

Email: Rolf.Reitzig@Siriuscom.com

DisciplinedAgileConsortium.org
DisciplinedAgileDelivery.com
Got Discipline?

DisciplinedAgileConsortium.org
DisciplinedAgileDelivery.com
ScottAmbler.com
DA Provides What You Need

• DA provides the flexibility, the detail, and the scale
• It’s been developed with you in mind
• When in doubt seek help
  – Books
  – Webinars
  – Blog
  – Consulting
Shuhari and Disciplined Agile Certification

At the *shu* stage you are beginning to learn the techniques and philosophies of disciplined agile development. Your goal is to build a strong foundation from which to build upon.

At the *ha* stage you reflect upon and question why disciplined agile strategies work, seeking to understand the range of strategies available to you and when they are best applied.

At the *ri* stage you seek to extend and improve upon disciplined agile techniques, sharing your learnings with others.