SIEMENS

Ingenuity for life

Ready for DevOps?

DevOps is neither a "social trend" nor a buzzword. It is a genuine business concern arising more and more from the daily pressures between your operations and information technology (IT) departments. This white paper introduces three different maturity levels in addressing the DevOps need, then provides a self-assessment questionnaire to help companies identify the maturity level of their existing DevOps approach. Also included are recommendations on how to reach a higher maturity level in merging together tools and methods between Development and Operations departments.

A white paper issued by: Siemens PLM Software

www.siemens.com/polarion

Contents

Is DevOps a real need?

It is fairly easy and often correct to ask whether a new marketing trend is just new, or if it originates from a genuine need. DevOps (the rising wave of tools and methods to put together development and operations departments) are based on solid arguments and not just marketing hysteria.

Some of the arguments are:

- Agile software development methods have increased the release frequency and pace, and as a result have enormously decreased the release preparation time
- Virtualization, cloud, mobile and mixed environments have also increased target system complexity
- Service-oriented architecture (SOA) and the integration of third-party services adds another degree of complexity into the mix
- So, while Agile seemed to solve most of the historical issues faced by the development labs in companies under time-to-market, continuous innovation and fast competition pressures, this "solution" actually moved the problem into operations. So the problem for the company remains unchanged.

What is DevOps?

DevOps has been defined as a movement, a set of principles, practices, concepts or methods. It is in fact an umbrella concept that refers to the software development methodology that improves the integration of silo activities through collaboration, communication and automation.

DevOps is the way to integrate development and operations to create a more collaborative and productive relationship between these teams whose operational tasks are interdependent.

In a traditional organization with separate departments for development, IT operations and quality assurance (QA), development methodologies (such as Agile software development) usually do not provide cross-departmental integration with IT support or QA.

The numerous steps involved, which cut across multiple domains such as development, QA, network engineering and operations makes the deployment of complex web-based and multitier applications highly error-prone, considering that many such steps involve manual intervention and routine tasks. The biggest pressure point for operations is the increased release frequency that is one of the most popular and appreciated practices common to every modern Agile development method. So while development seems to be able to survive the crazy time-to-market needs driving innovation, operations continues to struggle.

On the other hand, the adoption of Agile methods within development and their direct impact on operations with practices like continuous delivery have given some relief to operations: more and more Agile development teams manage the deployment.

To add to the complexity of operations, the use of public and hybrid cloud services is increasing, leading operations to manage application instances deployed to services instead of managing servers, backups and hardware faults.

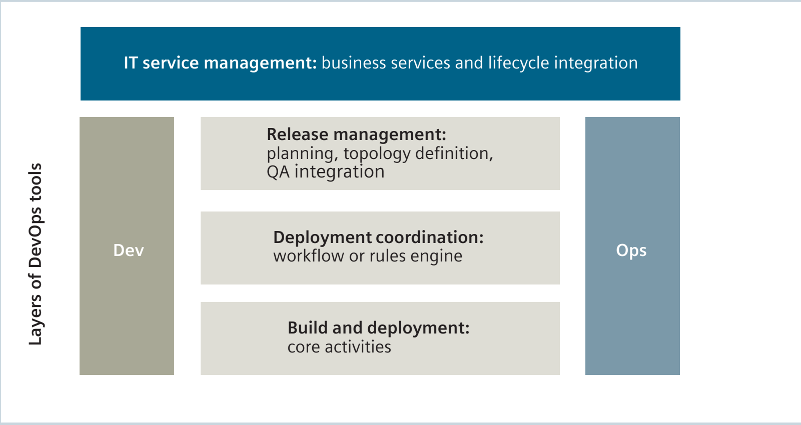
So the landscape of operations is changing even more rapidly than in the past, and some ops professionals have already been merged into Agile development teams. But in most cases, to increase operations efficiency and reduce the production risk it is critically necessary to improve overall collaboration and process automation between development and operations. This is, in a nutshell, the goal of DevOps.

We can say that DevOps is a way to put Agile practices into operations by adopting Agile values and practices. In a similar way that Agile has included testers as equals in the development process, build and ops are now no longer hidden in the basement corner. However, the newest members continue to use legacy tools that cannot keep up with their Agile teammates.

According to the well-respected analyst group Ovum, “DevOps is evolving to embrace business-facing IT services such as helpdesk and enterprise architecture through closer integration.”

> *“DevOps offers the opportunity to tackle these challenges and bring efficiencies into operations; improve collaboration between all stakeholders, adopting Agile values in working styles; and benefit from new automated solutions that target continuous delivery, release management, and deployment automation.”*

Michael Azoff,
Principal Analyst
Ovum


IT service management: business services and lifecycle integration
Layers of DevOps tools
Dev
Release management: planning, topology definition, QA integration
Deployment coordination: workflow or rules engine
Build and deployment: core activities
Ops

Source: Michael Azoff, “DevOps: Agile Operations and Continuous Delivery,” Ovum

Key takeaway

The paradigm is changing. Agile development and the increased use of public and hybrid cloud services are reshaping the operations center by removing inefficient and unsuitable work processes.

The adoption of Agile methodologies in development areas is resulting in higher-frequency deployments into production and therefore demands on operations are increasing. DevOps is designed to improve the effectiveness of an operations department, whose pace must be aligned with development’s frequent Agile releases. Release management and automation software solutions help organizations produce software products and services and address the urgency in getting the product to market on time.

DevOps approaches

DevOps is becoming a priority in most organizations today and there are several approaches to how you can execute a DevOps strategy. We have defined three different DevOps approaches:

Dinosaur approach (DinoOps) >

Dev and Ops are separated, with a basic communication channel.

Dev

- Provide ops with releases

Ops

- Provide dev with bugs and production failures

DinoOps best practice: define a release management process independent from the development process. The process must be easy and repeatable with predictable timing and results.

Mixed approach (MixOps) >

Dev and ops run separate processes with a common collaboration layer (bi-directional communication).

Dev

- Be involved in selecting the application stack
- Configure and deploy virtual or cloud servers (potentially)
- Deploy their applications
- Monitor application and system health
- Respond to applications problems as they arise

Ops

- Manage the hardware infrastructure
- Configure and monitor networking
- Enforce policies around backup, design release, security, compliance, change control, etc.
- Assist in monitoring the systems
- Manage active directory
- Asset tracking
- Other nonproduction, application-related tasks

Agile approach (AgileOps)

Dev and ops are the same team, sharing a single process (with ops backlog) orchestrated by a single product/platform.

Some consequences:

- User stories cover release and production needs
- There is a backlog owner for ops
- Every iteration covers deployments at least into test environment

DevOps backbone

In order to support MixOps, development and operations must share a common toolset, here termed DevOps backbone.

A DevOps backbone is the place where development and operations meet. It is the orchestra director of your DevOps, the shared platform to manage common processes and to ensure that development and operations are aligned. To some extent, it provides agility to the operations department and consciousness of operational needs into the development team.

The common processes managed by the DevOps backbone are:

- Collaboration (task and issue management, comments, approvals, message tracking, etc.)
- Version and change management: versioning, change requests, change management, collaborative coding, bug fixing and traceability management on scripts and code
- Software requirements management (requirements must also cover ops needs)
- QA (test cases, derived from requirements, must also include production tests; risk management encompasses production)

DevOps self-assessment

If you work in a company with separated development and IT operations departments you can assess the degree of maturity of your DevOps setup by answering a few questions pertaining to your specific role.

Yes / No

Management questions

1. Do your developers have visibility over operations progress? □
2. Does your IT have a consistent visibility over the actual development release schedule, including delays and cancellations? □
3. Do you involve operations in the early development stages so they can be aware of the future needs and start planning IT setup and deployment options? □
4. When something goes wrong is it clear who is in charge of it? □

Process questions

1. Do your business or user requirements include Software requirements about the production environment? □
2. Do you define test cases and acceptance criteria in the early phases of development that include production needs (examples: response time, number of concurrent users, uptime, etc.)? □
3. Do you plan development and deployment activities in the same chart? □

Toolset questions

1. Do your tools facilitate the communication between dev and ops (versus create a physical barrier between them, i.e. silos)? □
2. Do your tools support your process (versus mandate their process)? □
3. Do your dev and ops tools share the same workflow? □
4. Can the same workflow engine orchestrate your dev and ops tools? □
5. Are deployment scripts under version control? □
6. Can you track and manage changes to deployment scripts? □
7. Do you have traceability between development and deployment assets? □
8. Do you maintain all development and operational assets in a single asset database? □

Total Score: 0 x Yes

0–2 x Yes	3–6x x Yes	7–11 x Yes	12+ x Yes
Far from DevOps	In the initial DevOps adoption phase	Struggling with tools	DevOps hero

Far from DevOps

If you answered more than 12 “No,” then you are in the preliminary stages of DevOps maturity. You are not yet fully aware of the benefits DevOps can provide. If this describes your organization you will recognize that your operations are affected by the pressures described earlier in this paper.

Recommendation: consider moving your development towards an open and extensible application lifecycle management (ALM) platform. The recipe is to start with a MixOps approach, then merge teams into AgileOps after six to 12 months.

In the initial DevOps adoption phase

If you answered between 9 and 12 “No” answers, you are probably midway to creating a common process and your operations are getting closer to your development department.

Recommendation: Do not waste any more time and move straight to an AgileOps approach. In order to do that, start evaluating a unified platform for your dev and ops. If it’s not realistic for you, then you might follow the instructions below for those “struggling with tools.”

Struggling with tools

If you collected a significant number of “Yes” answers in the Management and Process questions, but you have more than 4 “No” answers in the Toolset questions, this may mean that you are already struggling with dev and ops tools misalignment.

Recommendation: Start to align dev and ops into an extensible unified platform. In case this is not possible, at least in the short term, you should adopt a MixOps approach. A DevOps backbone, as previously described, integrated or at least coupled with your development and operations tools, could be the right technical solution for you.

DevOps hero

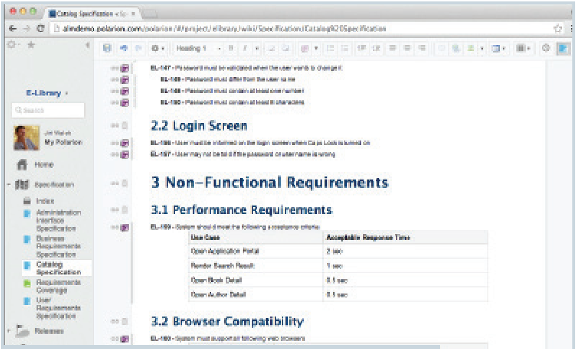
If you answered more than 12 “Yes,” please contact Siemens PLM Software. We are eager to hear how much you benefit from your DevOps approach and setup.

Your solution: Siemens PLM Software's Polarion ALM

Polarion® ALM is typically considered a powerful tool for development teams. An increasing number of our customers are using it more and more to manage their DevOps needs.

The following features explain why Polarion ALM customers are more successful with DevOps:

- Integration (actually Polarion ALM is a single tool, so there is nothing to integrate): Figure 1
- Information/asset uniqueness (single source, no replicas)
- Ability to store and version software, configuration files and deployment scripts: Figure 2
- Manage variants of software, configuration files and deployment scripts: Figure 3
- Define and re-use deployment workflows
- Provide content-driven workflow and process knowledge to tackle complex deployments
- Embedded QA features: Figure 4
- Govern (plan-execute-track) the change of deployed assets: Figure 5
- Ability to define and maintain patterns and templates in managing releases
- Embedded tools like Maven and Ant for many build and release activities: Figure 6
- Strong integrations with popular tools like Hudson and Jenkins
- Platform openness to any third-party release management or build tool


almdemo.polarion.com/polarion/#/project/elibrary/wiki/Specification/Catalog%20Specification

E-Library

EL-147 - Password must be validated when the user wants to change it

EL-148 - Password must differ from the user name

EL-149 - Password must contain at least one number

EL-150 - Password must contain at least 8 characters

2.2 Login Screen

EL-156 - User must be informed on the login screen when Caps Lock is turned on

EL-157 - User may not be told if the password or username is wrong

3 Non-Functional Requirements

3.1 Performance Requirements

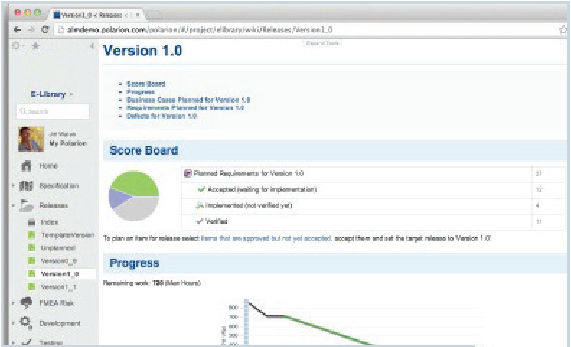
EL-159 - System should meet the following acceptance criteria:

Use Case	Acceptable Response Time
Open Application Portal	2 sec
Render Search Result	1 sec
Open Book Detail	0.5 sec
Open Author Detail	0.5 sec

3.2 Browser Compatibility

EL-160 - System must support all following web browsers

1) Live documents – traceable to source code


almdemo.polarion.com/polarion/#/project/elibrary/wiki/Releases/Version1_0

Version 1.0

- Score Board
- Progress
- Business Cases Planned for Version 1.0
- Requirements Planned for Version 1.0
- Defects for Version 1.0

Score Board

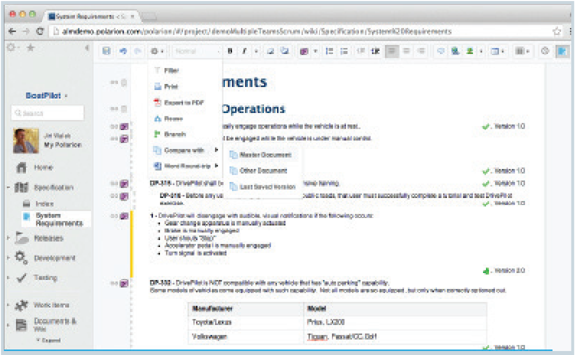
Planned Requirements for Version 1.0	27
✓ Accepted (waiting for implementation)	12
Implemented (not verified yet)	4
✓ Verified	11

To plan an item for release select items that are approved but not yet accepted, accept them and set the target release to 'Version 1.0'.

Progress

Remaining work: 720 (Man Hours)

2) Release management – flexible to any process


almdemo.polarion.com/polarion/#/project/demoMultipleTeamsScrum/wiki/Specification/System%20Requirements

BoatPilot

ments

Operations

DP-316 - DrivePilot shall be

DP-318 - Before any us

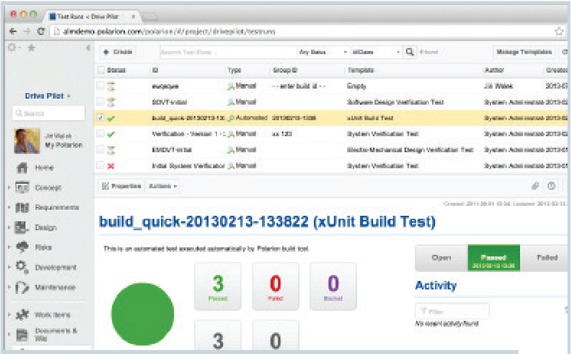
1 - DrivePilot will disengage with audible, visual notifications if the following occurs:

- Gear change apparatus is manually actuated
- Brake is manually engaged
- User shouts "Stop"
- Accelerator pedal is manually engaged
- Turn signal is activated

DP-332 - DrivePilot is NOT compatible with any vehicle that has "auto parking" capability. Some models of vehicles come equipped with such capability. Not all models are so equipped, but only when correctly optioned out.

Manufacturer	Model
Toyota/Lexus	Prius, LX200
Volkswagen	Tiguan, Passat/CC Ltd

3) Manage variants – no copy and paste to re-use any information


almdemo.polarion.com/polarion/#/project/drivepilot/testruns

Drive Pilot

Status	ID	Type	Group ID	Template	Author
	eucapae	Manual	-- enter build id --	Empty	Jiri Walek
	SDVT-initial	Manual		Software Design Verification Test	System Administrator
✓	build_quick-20130213-13	Automated	20130213-1338	xUnit Build Test	System Administrator
✓	Verification - Version 1	Manual	xx 123	System Verification Test	System Administrator
	EMDVT-initial	Manual		Electro-Mechanical Design Verification Test	System Administrator
✗	Initial System Verification	Manual		System Verification Test	System Administrator

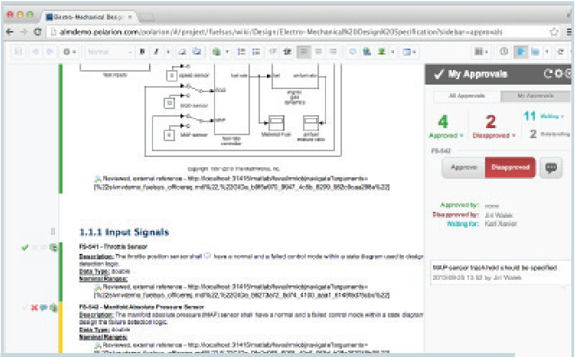
build_quick-20130213-133822 (xUnit Build Test)

This is an automated test executed automatically by Polarion build tool.

Activity

No recent activity found

4) QA features – for both manual and automated test


almdemo.polarion.com/polarion/#/project/fuelsys/wiki/Design/Electro-Mechanical%20Design%20Specification?sidebar=approvals

My Approvals

FS-542

Approved by: none
Disapproved by: Jiri Walek
Waiting for: Karl Xavier

MAP sensor thresholds should be specified
2013-09-25 13:52 by Jiri Walek

1.1.1 Input Signals

FS-541 - Throttle Sensor

Description: The throttle position sensor shall have a normal and a failed control mode within a state diagram used to design detection logic.

Data Type: double

Nominal Ranges:

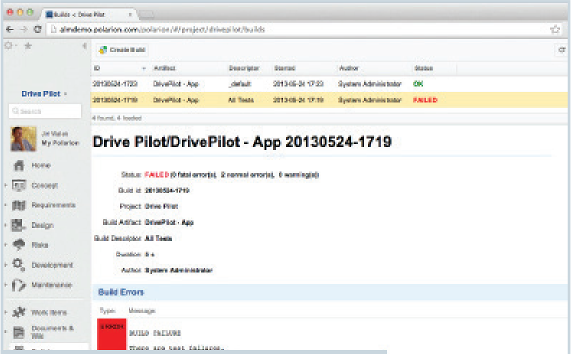
FS-542 - Manifold Absolute Pressure Sensor

Description: The manifold absolute pressure (MAP) sensor shall have a normal and a failed control mode within a state diagram design the failure detection logic.

Data Type: double

Nominal Ranges:

5) Managed collaboration – formal and informal approvals


almdemo.polarion.com/polarion/#/project/drivepilot/builds

Drive Pilot

Create Build

ID	Artifact	Descriptor	Started	Author	Status
20130524-1723	DrivePilot - App	_default	2013-05-24 17:23	System Administrator	OK
20130524-1719	DrivePilot - App	All Tests	2013-05-24 17:19	System Administrator	FAILED

4 found, 4 loaded

Drive Pilot/DrivePilot - App 20130524-1719

Status: FAILED (0 fatal error(s), 2 normal error(s), 0 warning(s))

Build id: 20130524-1719

Project: Drive Pilot

Build Artifact: DrivePilot - App

Build Descriptor: All Tests

Duration: 8 s

Author: System Administrator

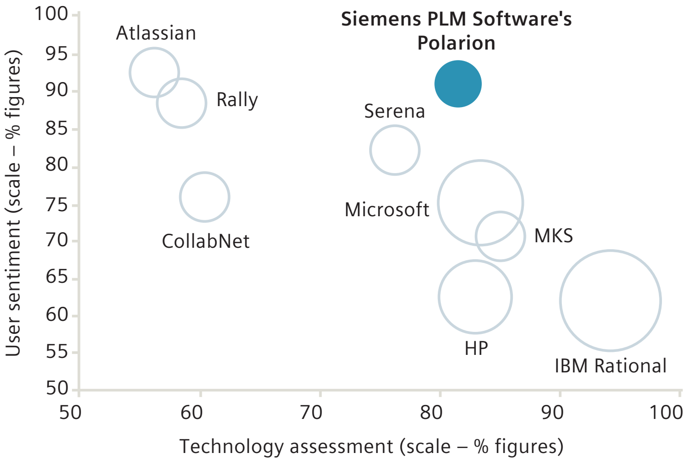
Build Errors

Type	Message
ERROR	BUILD FAILURE There are test failures.

6) Build management – based on maven

So is Polarion ALM the right choice for you to work with for DevOps? Are you in the initial DevOps phase or struggling with tools and the decision whether you should consider to move to a MixedOps or AgileOps approach? Polarion ALM, already the perfect toolset for development, is ready now to become your DevOps backbone or your single and unified AgileOps toolset. It's the right choice for you, unless you are already a Polarion DevOps hero.

ALM Decision Matrix



Source: Ovum/Siemens PLM Software

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

www.siemens.com/plm

55673-A11 8/16 F