



# Technology Radar

<http://www.thoughtworks.com/radar>

Prepared by the ThoughtWorks  
Technology Advisory Board

**January 2011**

**ThoughtWorks®**

## What's new?

**Since the last publication of the Technology Radar, these technology trends have become more prominent:**

- Focus on continuous delivery of software and methods to enable it
- Diversity of and depth in cloud offerings
- Using basic web technologies in more effective and efficient ways

## Introduction

The ThoughtWorks Technology Advisory Board is a group of senior technology leaders within ThoughtWorks. They produce the ThoughtWorks Technology Radar to help decision makers understand emerging technologies and trends that affect the market today. This group meets regularly to discuss the global technology strategy for ThoughtWorks and the technology trends that significantly impact our industry.

The Technology Radar captures the output of these discussions in a format that provides value to a wide range of stakeholders, from CIOs to enterprise developers. The content provided in this document is kept at a summary level, leaving it up to the reader to pursue more detailed knowledge when needed. The goal of the radar is conciseness, so that its target audience understands it quickly. The radar is graphical in nature, grouping items into techniques, tools, languages and platforms. Some radar items could appear in multiple quadrants, but we mapped them to the quadrant that seemed most appropriate. We further group these items in four rings to reflect our current position on them. The rings are

- Hold: Proceed with caution.
- Assess: Worth exploring with the goal of understanding how it will affect your enterprise.

- Trial: Worth pursuing. It is important to understand how to build up this capability. Enterprises should try this technology on a project that can handle the risk.
- Adopt: Industry has finished trial and found proper patterns of usage, or we feel strongly that the industry should be adopting it now, rather than going through a gradual adoption.

As we look at each quadrant in detail, we show the movement that each item has taken since the last publication of the radar. New items are represented as triangles, while items that were on the last radar are represented as circles. Items that have not moved in two publications of the radar fade and are no longer displayed unless something significant happens.

## Contributors

The ThoughtWorks Technology Advisory Board is comprised of

Rebecca Parsons (CTO)  
 Martin Fowler (Chief Scientist)  
 Nick Hines (CTO Innovation)  
 Evan Bottcher  
 Graham Brooks  
 Ian Cartwright  
 Erik Doernenburg

Jim Fischer  
 Neal Ford  
 Ajey Gore  
 Wendy Istvanick  
 Mike Mason  
 Cyndi Mitchell

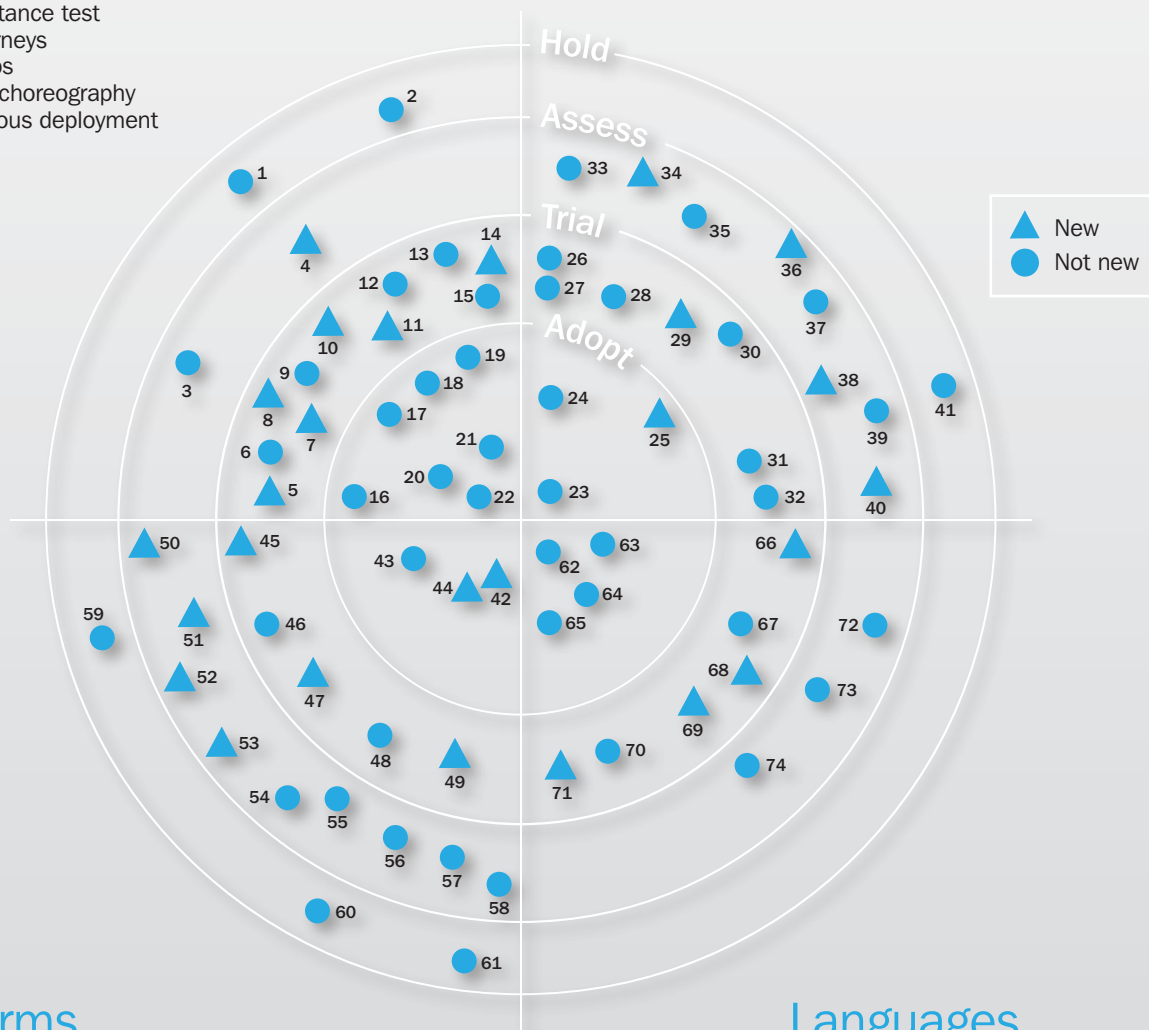
David Rice  
 Ian Robinson  
 Pramod Sadalage  
 Samir Seth  
 Scott Shaw  
 Hao Xu  
 Jeff Norris

## Techniques

1. Database based integration
2. Scrum certification
3. Real-time business intelligence
4. ▲ Smart Systems
5. ▲ Progressive Enhancement
6. Automation of technical tests
7. ▲ Automate database deployment
8. ▲ Concurrency abstractions and patterns
9. Capability modelling
10. ▲ Acceptance test of journeys
11. ▲ DevOps
12. Service choreography
13. Continuous deployment
14. ▲ Categorization & prioritization of technical debt
15. Evolutionary architecture
16. Coding architects
17. Visualisation and metrics
18. Web as platform
19. Emergent design
20. Evolutionary database
21. Platform roadmaps
22. Build pipelines

## Tools

23. Subversion
24. Squid
25. ▲ Infrastructure as code
26. Apache camel
27. Message buses without smarts
28. Next gen test tools
29. ▲ Splunk
30. NoSQL
31. Mercurial
32. Git
33. Cross mobile platforms
34. ▲ Deltacloud
35. Github
36. ▲ Vagrant
37. Restfulie
38. ▲ WCF HTTP
39. RDF triple stores
40. ▲ API management services
41. ESB



## Platforms

42. ▲ KVM
43. Android
44. ▲ Atom
45. ▲ Heroku
46. Facebook as business platform
47. ▲ iPad
48. EC2 & S3
49. ▲ Mobile Web
50. ▲ GPGPU
51. ▲ Node.js
52. ▲ vFabric
53. ▲ OpenStack
54. Application appliances
55. RDFa
56. OAuth
57. App containers
58. Azure
59. WS-\* beyond basic profile
60. GWT
61. RIA

## Languages

62. Ruby
63. C# 4.0
64. JRuby
65. JavaScript as a first class language
66. ▲ SASS, SCSS, and LESS
67. DSL's
68. ▲ HAML
69. ▲ Scala
70. Groovy
71. ▲ HTML 5
72. Java language end of life
73. F#
74. Clojure

## Techniques

Business are running 24x7, but the information needed to make business decisions is being provided by outdated methods using ETL jobs in batch mode. The batch window is shrinking as businesses move to global markets and are open for longer durations. The data provided by these jobs is out of date by the time the business needs to make a decision. There is substantial value in taking the event as it happens in the transactional system and feeding it to the data warehouse so that the business can get near **real-time business intelligence**.

Smart Phones with a GPS, cameras and a screen are but one example of **smart systems** which are proliferating around us, fusing the real and the digital world. Augmented reality apps like Google Goggles, geolocation services & smart grids are just some of the possible applications.

The **DevOps** movement continues to gain traction as people pay more attention to the often-broken relationship between development and operations. DevOps promotes closer collaboration and joint responsibility between development and operations. DevOps applies agile practices to operations processes such as provisioning, change management and production monitoring and also brings production-like thinking, tools and environments to development. DevOps is a key underpinning for organizations wanting to achieve continuous delivery of application releases into production.

When moving to continuous delivery, deployment of database changes should also be automated so that the application release that relies on those changes does not have to wait for manual deployment of the database changes. **Automated database deployment** ensures that the full cycle of deploying application and database changes is automated.

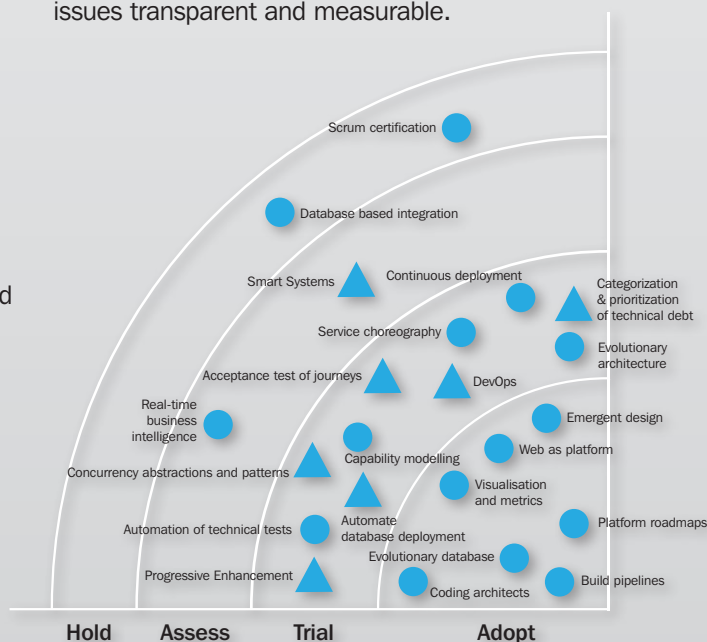
Many teams focus acceptance testing at the story level, but this can lead to a large number of hard to maintain tests, and a focus on completing individual stories instead of coherent functionality needed to go live. A more holistic approach is to group user stories into journeys for which we create acceptance tests. Journeys through a system are a set of user interactions that provide value for both users and the business. At the outset a journey acceptance test will cover only one step, but as stories are completed the journey is expanded to encompass

each stage in the user's progress. Once the **acceptance test of journeys** passes, this tells us we have delivered real value.

**Progressive enhancement** is a web design strategy that uses layers of web technologies to build a compelling user experience. Progressive enhancement emphasizes accessibility, allowing access to the basic content using any browser. This strategy enables more content to be cached which improves overall system performance and scalability.

Concurrency is a difficult problem and increasingly difficult to avoid. Hardware isn't getting faster but multicore platforms are becoming the norm, with even mobile phones containing two or more processors. **Concurrency abstractions and patterns** -- which are not new, but less widely known -- are helping address many of the challenges seen in this space. In particular the models seen in Clojure, Erlang, Retlang and Event Patterns offer a more testable and reliable approach than the better known threads, locks and semaphores.

Technical debt is a powerful and useful metaphor for dealing with the compromises we make when building software. Unfortunately it has become a catch-all term for many different kinds of issues and problems, leading to confusion and "devaluation" of the term. A very useful approach for dealing with this is **categorization of technical debt**, assigning value and prioritizing debt payback in an analogous way to user stories. This helps the team focus on the most important areas and keeps issues transparent and measurable.



## Tools

The large number of hosts and devices in a modern datacenter or cloud deployment have made manually installing and configuring infrastructure unwise.

**Infrastructure as code** is an approach whereby infrastructure configuration is scripted or described by files that are stored in version control, and changes are pushed out to the datacenter in a controlled manner. This parallels the discipline of source control and build promotion used in software development, hence 'as code'. The two front-running open source tools for infrastructure automation are Chef and Puppet. They both use a textual DSL to script automation. Using this approach provides consistent and repeatable environment changes, reducing the manual effort involved, especially in troubleshooting environmental differences.

The tool **Vagrant** makes it simple for teams to distribute virtualized development environments constructed using version-control friendly descriptors. Vagrant helps eliminate environmental differences in development and reduce troubleshooting waste.

Every Infrastructure as a Service (IaaS) cloud offering provides their own API for performing common tasks. **Deltacloud** aims to abstract those APIs and provide a RESTful interface for performing common cloud management functions, making it possible to migrate virtual infrastructure between clouds.

Exposing APIs to a large developer community creates nonfunctional requirements that are often the same from one business to another. Key management, authentication, access control, traffic management, caching, tracking and reporting are often implemented as commodity functions that can be reused across applications and businesses without modification. Some service providers have spotted this trend and are offering **API management** via software as a service. Prominent vendors in this space include Mashery and Apigee, who both offer the option of hosting the service on a customer's own infrastructure. API management services may also be interesting to enterprise customers who are using Web as platform techniques for their internal SOA, providing a lighter weight alternative to traditional SOA management tools.

Microsoft's **WCF HTTP** API, though currently in the early stages of development, is evolving rapidly, and we're already impressed by its support for HTTP

primitives as well as idioms such as content negotiation and conditional requests. The API encourages the development of highly testable solutions with a clear separation of concerns. What is of particular interest to us is the way in which the project is being developed in Codeplex's open source community. The ability for the community to steer the development of this part of the .NET platform merits this project's early inclusion in the radar. While the license does allow for using the library in production today, given the current volatility of the API, we caution against taking a dependency at this early stage; many of the features we've admired in recent releases, such as its use of an attribute-light programming model based on convention over configuration, may not make it into the first version.

Application logs are both a blessing and a curse. They are comforting to have when a production issue arises, but actually digging out the data we need usually requires cobbling together scripts written in tools such as AWK and sed. **Splunk** is an elegant solution that quickly analyzes many standard log file formats like IIS, Log4J and syslog, and is extensible to custom formats. It indexes files, statically or in real time, to generate canned or custom reports. If the raw log fields do not provide what you need, simply use a regular expression, either inline or to define a new field, to get the desired level of detail. Splunk's full power is difficult to describe, so we recommend downloading and trying it.



## Languages

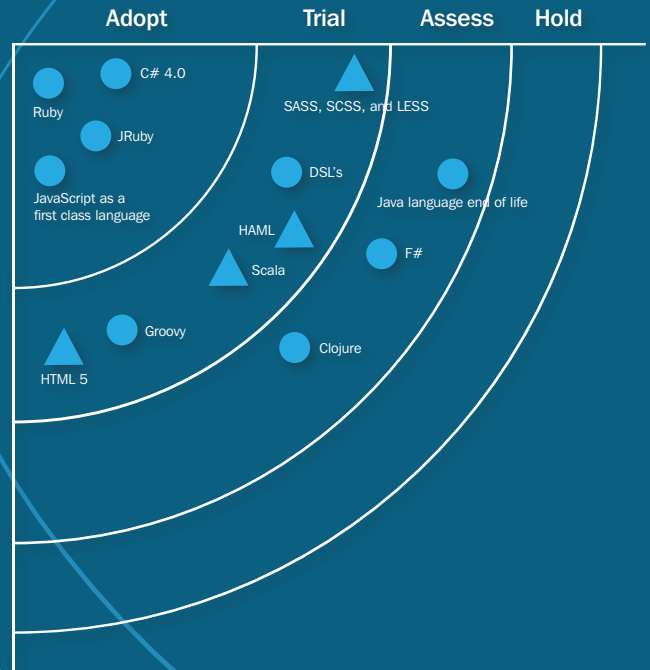
The purchase of Sun, and thus their Java assets, by Oracle introduced uncertainty regarding the **future of Java**. This uncertainty continues despite the recent announcements of Oracle's Java roadmap, which had both encouraging and worrying aspects. As a result we continue to highlight the issue. We recommend monitoring the situation rather than any immediate actions to move off the platform.

In the previous radar we had two JVM-based functional programming languages, Clojure and **Scala**, in our Assess category. We had expressed a slight preference for Clojure because it is the smaller and more focused language. Since the last radar we have realized that the wider applicability of Scala makes it more approachable for enterprise developers, and we have witnessed great successes in the adoption of Scala. Consequently we have moved Scala into our Trial category. Pay careful attention to the idiomatic use of Scala if it is introduced to a new team to avoid "Java without semicolons" or Perl styles.

CSS has been an extremely popular and effective DSL for styling web pages. It does, however, have some annoying limitations which have led to a number of languages that build on CSS to make it easier to write and modify. We've had good experiences with **SASS, SCSS, and LESS**.

**HAML** is a language that allows you to use indentation to lay out the structure of HTML. While not a general replacement for HTML, it is effective for focusing on the hierarchical structure of tags.

**Domain-Specific Languages** is an old technique that we think is significantly under-used. We hope that the publication of Martin Fowler's latest book will encourage more people to utilize them.





## Platforms

**OpenStack** is a new cloud operating system that promises a complete open-source solution. OpenStack is a fabric cloud controller which leverages existing virtualization technologies such as KVM and will integrate with other virtualisation tools such as Xen and OpenVZ. Currently under heavy development, OpenStack is expected to provide a stable production-ready solution by the end of Q2 2011.

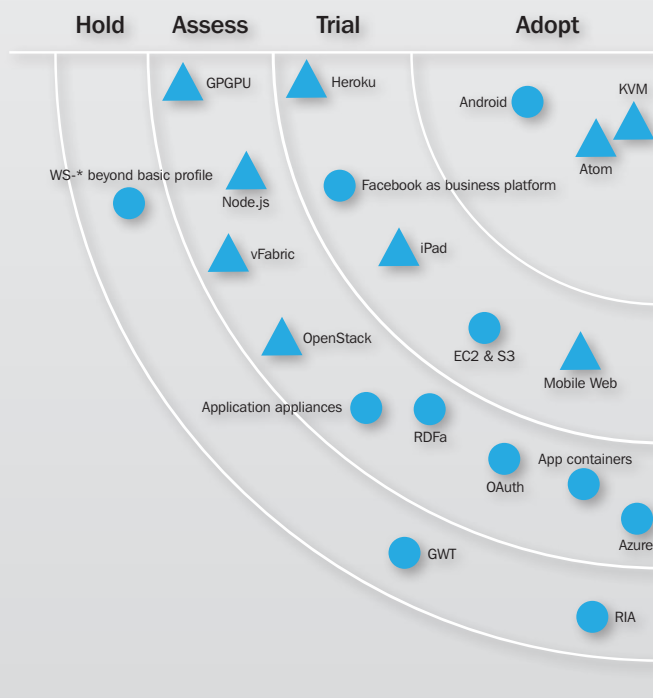
**vFabric** is a new Platform as a Service (PaaS) offering from VMWare. Based on enhanced versions of open-source web and messaging platforms Tomcat, Apache, and RabbitMQ, vFabric aims to deliver a Java based PaaS on a variety of cloud platforms. Currently supported platforms include VMForce, a collaboration between VMWare and force.com, Google App Engine and Amazon EC2. The addition of the GemFire in-memory distributed data management platform and Hyperic monitoring and management tool make vFabric an interesting set of technologies for Java developers looking to move to the cloud.

For many years JavaScript was predominantly used as a client side Web programming language, but a lightweight language such as JavaScript can easily be embedded in different environments, including the server side. **Node.js** allows developers to write applications in JavaScript on both client and server sides. Since most servers spend the majority of their time waiting for I/O operations, Node.js' event driven non-blocking architecture is very efficient. Unlike thread-based solutions, Node.js does not need to wait for I/O operations to complete while processing incoming requests, making it a good choice when implementing high performance services.

**Heroku** is a beautifully simple Platform as a Service (PaaS) for Rack-compatible frameworks such as Ruby on Rails. In contrast to similar offerings for other languages, which often limit development to a programming model specific to the service platform, Heroku uses the standard Rails stack and even allows deployment with a plain Git push. Heroku was recently acquired by Salesforce.com and so has significant backing behind their service.

**Mobile Web** was in our Assess category on previous radars, but we've moved it into Trial in recognition of the fact that the Web is increasingly consumed through iPhone, Android and tablet devices. Many devices can render a fair approximation of a desktop browser experience, but to present the user with a truly optimal experience we recommend adapting a site to the screen size, display and control behaviors particular to the device. Techniques such as progressive enhancement can allow a single site to adapt to both desktop and mobile browsers.

Large format mobile devices, such as the Apple iPad and Amazon Kindle, provide a new model of ubiquitous computing. Their long battery life, simple interfaces and easy connectivity have the potential to change the way we interact with computers. Apple's new user interfaces discard the familiar desktop metaphors of files and folders that have been standard since the introduction of the Macintosh in 1984.



## References

Apache Deltacloud project <http://incubator.apache.org/deltacloud>  
Automated database deployment whitepaper <http://bit.ly/automateDb>  
Coming War Over the Future of Java <http://bit.ly/javaFuture>  
Concurrency patterns [http://en.wikipedia.org/wiki/Concurrency\\_pattern](http://en.wikipedia.org/wiki/Concurrency_pattern)  
HAML <http://haml-lang.com>  
Heroku <http://heroku.com>  
HTML 5 Rocks! <http://slides.html5rocks.com>  
Implications of Infrastructure as Code <http://bit.ly/InfraAsCode>  
Martin Fowler's Bliki: Domain Specific Languages <http://martinfowler.com/bliki/DomainSpecificLanguage.html>  
Mashery API management <http://mashery.com>  
Node.js <http://nodejs.org>  
OpenStack <http://openstack.org>  
Progressive Enhancement [http://en.wikipedia.org/wiki/Progressive\\_enhancement](http://en.wikipedia.org/wiki/Progressive_enhancement)  
Realtime business intelligence [http://en.wikipedia.org/wiki/Real-time\\_business\\_intelligence](http://en.wikipedia.org/wiki/Real-time_business_intelligence)  
SASS - Syntactically Awesome Stylesheets <http://sass-lang.com>  
Scala Programming Language <http://scala-lang.org/node/25>  
Smart Systems - Real & digital worlds converging <http://economist.com/node/17388368>  
Splunk <http://splunk.com>  
Technical Debt <http://martinfowler.com/bliki/TechnicalDebt.html>  
Vagrant virtualized development environments <http://vagrantup.com>  
vFabric Cloud Application Platform <http://springsource.com/products>  
WCF Codeplex project <http://wcf.codeplex.com>  
What is this DevOps thing, anyway? <http://bit.ly/whatisdevops>

## ThoughtWorks is a global IT consultancy

We deliver custom applications and provide consulting grounded in reality; we help organizations become efficient through Agile and Lean practices and principles. By hiring exceptional people, we can solve our clients' biggest and most pressing problems. All of our services are offered both on and offshore, and are delivered with pride and passion.