APRIL '16

Our thoughts on the technology and trends that are shaping the future

Here are the themes highlighted in this edition:

Some of the most influential software appearing on our radar comes from companies whose first mandate isn't to create software tools. Several of our radar entries come from Facebook, not considered a traditional software development toolmaker. Unlike in the past, today many companies open source their important software assets—to attract new recruits and credentialize themselves. This creates a virtuous feedback loop: Innovative open source attracts good developers who are in turn more likely to innovate. As a side effect, these companies' frameworks and libraries are some of the most influential in the industry. This represents a big shift in the software development ecosystem and is further proof of the efficacy of open source software ... in the right context (our advice about Web Scale Envy still stands).

Many large organizations see the Cloud and Platform as a Service (PaaS) as an obvious way to standardize infrastructure, ease deployment and operations, and make developers more productive. But it's still early days, the definition of PaaS remains nebulous, and many PaaS approaches are incomplete or suffer from the immaturity of supporting frameworks and tools. Some PaaS solutions make it harder to do things more easily done with plain Infrastructure as a Service (IaaS), such as using a custom Service Locator or complex network topology, and the jury is still out on whether a "Containers as a Service" approach will provide similar value with more flexibility. We see many companies implementing an off-the-shelf PaaS or gradually building their own, with varying degrees of success. We suspect that any PaaS built today will not be an end state but rather part of an evolutionary path. Enterprise migration to Cloud and PaaS, while bringing many benefits, has difficulties and challenges, particularly around overall pipeline design and tooling. Consumers of these technologies should seek the inflection point that indicates "ready for prime time" for their context and should avoid coupling too tightly to the implementation details of their PaaS.

Containerization, and Docker in particular, has proven hugely beneficial as an application-management technique, rationalizing deployment between environments and simplifying the "it works here but not there" class of problems. We see a significant amount of energy focused on using Docker—and, particularly, the ecosystem surrounding it—beyond dev/test and all the way into production. Docker containers are used as the "unit of scaling" for many PaaS and "data center OS" platforms, giving Docker even more momentum. As it matures as both a development and production environment, people are paying more attention to containerization, its side effects and its implications.

Reactive programming—where components react to changes in data that are propagated to them rather than use imperative wiring—has become extremely popular, with reactive extensions available in almost all programming languages. User interfaces, in particular, are commonly written in a reactive style, and many ecosystems are settling on this paradigm. While we like the pattern, overuse of event-based systems complicates program logic, making it difficult to understand; developers should use this style of programming judiciously. It is certainly popular: We added a significant number of reactive frameworks and supporting tools on this Radar.

The Technology Radar is prepared by the ThoughtWorks Technology Advisory Board, comprised of:

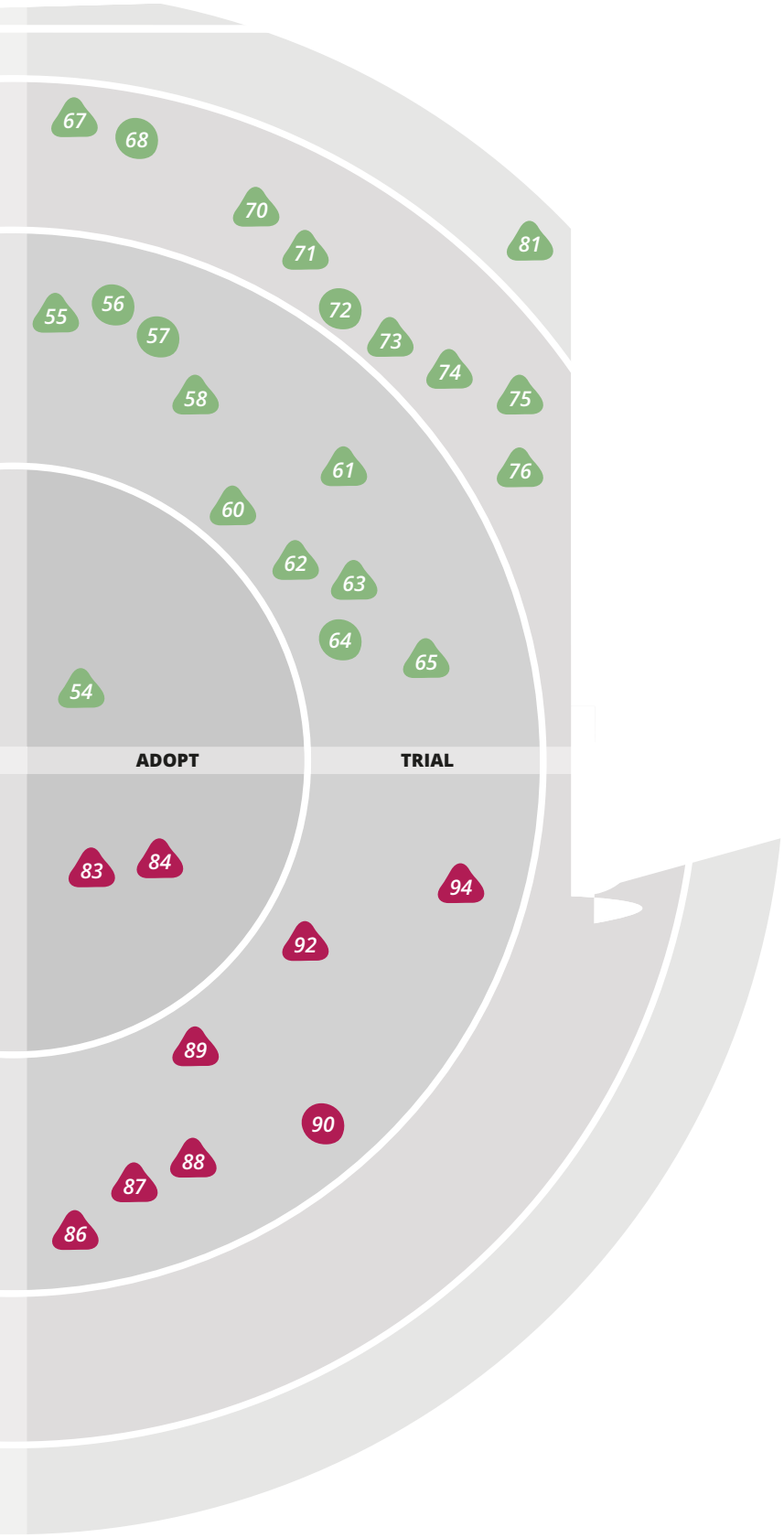
Rebecca Parsons (CTO)	Dave Elliman	Ian Cartwright	Rachel Laycock
Martin Fowler(Chief Scientist)	Erik Doernenburg	James Lewis	Sam Newman
Anne J Simmons	Evan Bottcher	Jonny LeRoy	Scott Shaw
Badri Janakiraman	Fausto de la Torre	Mike Mason	Srihari Srinivasan
Brain Leke	Hao Xu	Neal Ford	Thiyagu Palanisamy

ThoughtWorkers are passionate about technology. We build it, research it, test it, open source it, write about it, and constantly aim to improve it – for everyone. Our mission is to champion software excellence and revolutionize IT. We create and share the ThoughtWorks Technology Radar in support of that mission. The ThoughtWorks Technology Advisory Board, a group of senior technology leaders in ThoughtWorks, creates the radar. They meet regularly to discuss the global technology strategy for ThoughtWorks and the technology trends that signi fcantly impact our industry.

The radar captures the output of the Technology Advisory Board's discussions in a format that provides value to a wide range of stakeholders, from CIOs to developers. The content is intended as a concise summary. We enc0a1tte you to explore these technologies for more detail. The radar is graphical in nature, grouping items into techniques, tools, platforms, and languages & frameworks. When radar items could appear in multiple quadrants, we chose the one that seemed most appropriate. We further group these items in four rings to re fect our current position on them. The rings are:

Items that are new or have had signi fcant changes since the last radar are represented as triangles, while items that have not moved are represented as circles. We are interested in far more items than we can reasonably f t into a document this size, so we fade many items from the last radar to make room for the new items. Fading an item does not mean that we no longer care about it.

For more background on the radar, see thoughtworks.com/radar/faq



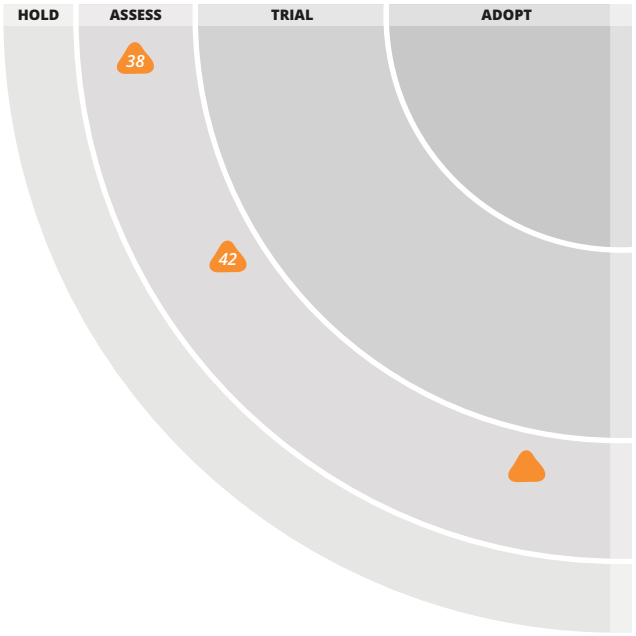
With the number of high-profile security breaches in the past months, software development teams no longer need convincing that they must place an emphasis on writing secure software and dealing with their users' data in a responsible way. The teams face a steep learning curve, though, and the vast number of potential threats—ranging from organized crime and government spying to teenagers who attack systems "for the lulz"—can be overwhelming. _____ provides a set of techniques that help you identify and classify potential threats early in the development process. It is important to understand that it is only part of a strategy to stay ahead of threats. When used in conjunction with techniques such as establishing cross-functional security requirements to address common risks in the technologies a project uses and using automated security scanners, threat modeling can be a powerful asset.

The use of

continued

While we've long understood the value of Big Data to better understand how people interact with us, we've noticed an alarming trend of : organizations using complex tools to handle "not-really-that-big" Data. Distributed map-reduce algorithms are a handy technique for large data sets, but many data sets we see could easily fit in a single-node relational or graph database. Even if you do have more data than that, usually the best thing to do is to first pick out the data you need, which can often then be processed on such a single node. So we urge that before you spin up your clusters, take a realistic assessment of what you need to process, and if it fits—maybe in RAM—use the simple option.

We remain excited about _____ as it evolves from a tool to a complex platform of technologies. Development teams love Docker, as the Docker image format makes it easier to achieve parity between development and production, making for reliable deployments. It is a natural fit in a microservices-style application as a packaging mechanism for self-contained services. On the operational front, Docker support in monitoring tools (Sensu, Prometheus, cAdvisor, etc.), orchestration tools (Kubernetes, Marathon, etc.) and deployment-automation tools reflect the growing maturity of the platform and its readiness for production use. A word of caution, though: There is a prevalent view of Docker and Linux containers in general as being “lightweight virtualization,” but we



continued

The PaaS space has seen a lot of movement since we last mentioned Cloud Foundry in 2012. While there are various distributions of the open source core, we have been impressed by the offering and ecosystem assembled as _____. While we expect continued convergence between the unstructured approach (Docker, Mesos, Kubernetes, etc.) and the more structured and opinionated buildpack style offered by Cloud Foundry and others, we see real benefit for organizations that are willing to accept the constraints and rate of evolution to adopt a PaaS. Of particular interest is the speed of development that comes from the simplification and standardization of the interaction between development teams and platform operations.

The emerging Containers as a Service (CaaS) space is seeing a lot of movement and provides a useful option between basic IaaS (Infrastructure as a Service) and more opinionated PaaS (Platform as a Service). While _____ creates less noise than some other players, we have enjoyed the simplicity that it brings to running Docker containers in production. It can run stand-alone as a full solution or in conjunction with tools like Kubernetes.

_____ is Amazon's offering enabling developers to expose API services to Internet clients, offering the usual API gateway features like traffic management, monitoring, authentication and authorization. Our teams have been using this service to front other AWS capabilities like AWS Lambda as part of serverless architectures. We continue to monitor for the challenges presented by over-ambitious API gateways, but at this stage Amazon's offering appears to be lightweight enough to avoid those problems.

While many deployments of smart devices rely on Wi-Fi connectivity, we have been seeing success with networks that don't necessitate a hub or gateway. With better energy usage than Wi-Fi and better smartphone adoption than ZigBee, Bluetooth LE deployed as a self-healing mesh provides interesting new approaches for connecting local device-area networks. We are still waiting for the formal approach to emerge from the Bluetooth SIG but have already had successful deployments. We particularly like the lack of infrastructure required to stand up a decentralized network but still retain the option to "progressively enhance" the system with the addition of a gateway and cloud services.

Defect is an open source service protecting NGOs, activist and independent media companies from DDoS attacks. Similar to a commercial CDN, it uses distributed reverse-proxy caching and also hides your server IP addresses and blocks public access to admin URLs. Particular effort is put in to combat the botnets typically used for extrajudicial censoring of independent voices.

Our growing ranks of hardware hackers have been excited by the _____ Wi-Fi microcontroller. Rather than a specific technology innovation, it is the combination of low price point and small form factor that has sparked an infection point in people's thinking about what is now feasible to achieve with custom hardware devices. Its main characteristics are: Wi-Fi capabilities (it can act as station, access point or a combination of both), low power, open hardware, Arduino SDK programmability, Lua programmability, huge community support and low cost compared with other IoT modules.

As Moore's Law predicts, we continue to increase the capacity of computer systems and reduce their cost, and so new processing techniques become possible that only a few years ago would have seemed out of reach. One of these techniques is the in-memory database: Instead of using slow disks or relatively slow SSDs to store data, we can keep it in memory for high performance. One such in-memory database, _____, is making waves because it is horizontally scalable across a cluster and provides a familiar SQL-based query language. MemSQL also connects to Spark for analytics against real-time data, rather than stale data in a warehouse.

HashiCorp continues to turn out interesting software. The latest to catch our attention is _____, which is competing in the ever-more-populated scheduler arena. Major selling points include not just being limited to containerized workloads, and operating in multi–data center / multiregion deployments.

_____ is a database designed for use on mobile devices, with its own persistence engine to achieve high performance. Realm is marketed as a replacement for SQLite and Core Data, and our teams have enjoyed using it. Note that migrations are not quite as straightforward as the Realm documentation would have you believe. Still, Realm has us excited, and we suggest you take a look.

continued

For people who want the benefit of cloud-based collaboration tools but don't want to inadvertently "become the product" of a major cloud provider, _____ provides an interesting open source alternative with the potential for self-hosting. Of particular interest is the isolation approach, whereby containerization is applied per document rather than per application, and syscall whitelisting is added to further secure the sandbox.

Google's _____ is an open source machine-learning platform that can be used for everything from research through to production and will run on hardware from a mobile CPU all the way to a large GPU compute cluster. It's an important platform because it makes implementing deep-learning algorithms much more accessible and convenient. Despite the hype, though, TensorFlow isn't really anything new algorithmically: All of these techniques have been available in the public domain via academia for some time. It's also important to realize that most businesses are not yet doing even basic predictive analytics and that jumping to deep learning likely won't help make sense of most data sets. For those who do have the right problem and data set, however, TensorFlow is a useful toolkit.

We have moved _____, the service-discovery tool supporting both DNS- and HTTP-based discovery mechanisms, into Adopt. It goes beyond other discovery tools by providing customizable health checks for registered services, ensuring that unhealthy instances are marked accordingly. More tools have emerged to work with Consul to make it even more powerful. Consul Template enables configuration files to be populated with information from Consul, making things like client-side load balancing using mod_proxy much easier. In the world of Docker, registrator can automatically register Docker containers as they appear with Consul with extremely little effort, making it much easier to manage container-based setups. You should still think long and hard about whether you need a tool like this or whether something simpler will do, but if you decide you need service discovery, you won't go far wrong with Consul.

Many organizations are now looking closely at new data architectures that capture information as immutable sequences of events at scale. _____ continues to build momentum as an open source messaging framework that provides a solution for publishing ordered event feeds to large numbers of independent, lightweight consumers. Configuring Kafka is nontrivial, but our teams are reporting positive experiences with the framework.

continued

In a world full of libraries and tools that simplify the life of many software developers, deficiencies in their security have become visible and have increased the vulnerability surface in the applications that use them. _____ automatically identifies potential security problems in the code, checking if there are any known publicly disclosed vulnerabilities, then using methods to constantly update the database of public vulnerabilities. Dependency-Check has some interfaces and plugins to automate this verification in Java and .NET (which we have used successfully) as well as Ruby, Node.js and Python.

In the past we have included automated Provisioning Testing as a recommended technique, and in this issue we highlight _____ as a popular tool for implementing those tests. Although this tool is not new, we are seeing its use become more common as more cross-functional delivery teams take on responsibility for infrastructure provisioning. Serverspec is built on the Ruby library RSpec and comes with a comprehensive set of helpers for asserting that server configuration is correct.

_____ has solidified itself as our go-to JavaScript module bundler. With its ever-growing list of loaders, it provides a single dependency tree for all your static assets, allowing flexible manipulation of JavaScript, CSS, etc. and minimizing what needs to be sent to the browser and when. Of particular relevance is the smooth integration among AMD, CommonJS and ES6 modules and how it has enabled teams to work in ES6 and seamlessly transpile (using Babel) to earlier versions for browser compatibility. Many of our teams also value Browserify, which covers a similar space but is more focused on making Node.js modules available for client-side use.

Development on _____ has continued apace, and since the middle of 2015 it has moved to the openzipkin/zipkin organization at GitHub. There are now bindings for Python, Go, Java, Ruby, Scala and C#; and there are Docker images available for those wanting to get started quickly. We still like this tool. There is an active and growing community around usage of it, and implementation is getting easier. If you need a way of measuring the end-to-end latency of many logical requests, Zipkin continues to be a strong choice.

_____ is a new-generation platform for scalable distributed batch and stream processing. At its core is a streaming data-flow engine. It also supports tabular (SQL-like), graph-processing and machine-learning operations. Apache Flink stands out with feature-rich capabilities for stream processing: event time, rich streaming window operations, fault tolerance and exactly-once semantics. While it hasn't reached version 1.0, it has raised significant community interest due to innovations in stream processing, memory handling, state management and simplicity of configuration.

Attackers continue to use automated software to crawl public GitHub repositories to find AWS credentials and spin up EC2 instances to mine Bitcoins or for other nefarious purposes. Although adoption of tools like git-crypt and Blackbox to safely store secrets such as passwords and access tokens in code repositories is increasing, it is still all too common that secrets are stored unprotected. It is also not uncommon to see project secrets accidentally checked in to developers' personal repositories. _____ can help minimize the damage of exposing secrets. It scans an organization's GitHub repositories, flagging all files that might contain sensitive information that shouldn't have been pushed to the repository. The current release of the tool has some limitations: It can only be used to scan public GitHub organizations and their members, it doesn't inspect the contents of files, it doesn't review the entire commit history, and it fully scans all repositories each time it is run. Despite these limitations, it can be a helpful reactive tool to help alert teams before it is too late. It should be considered a complementary approach to a proactive tool such as Talisman.

We had our collective minds blown by a little JavaScript command-line refactoring tool called _____. Providing a rich set of selectors and operating against the abstract syntax tree, it is leagues ahead of fiddling with sed and grep. A useful addition to the toolkit in our ongoing quest to treat JavaScript as a first-class language.

Having a way to securely manage secrets is increasingly becoming a huge project issue. The old idea of just having a file with secrets or environment variables is becoming hard to manage, especially in environments with multiple applications like microservices or microcontainer environments, where the applications need to access a multitude of secrets. _____ is a promising tool that tries to solve the problem by providing mechanisms for securely accessing secrets through an unified interface. It has some features that make life easier, such as encryption and automatically generating secrets for known tools, among others.

With the growth in usage of NoSQL data stores, and the growth in popularity of polyglot approaches to persistence, teams now have many choices when it comes to storing their data. While this has brought many advantages, product behavior with faky networks can introduce subtle (and not so subtle) issues that are often not well understood, even in some cases by the product developers themselves. The _____ toolkit and accompanying blog have become the de-facto reference for anyone looking to understand how different database and queuing technologies react under adverse conditions. Crucially, the approach to testing, which includes clients in the transactions, shines a spotlight on possible failure modes for many teams building microservices.

_____ provides teams with a way to define Continuous Delivery pipelines in Clojure. This brings the benefts of Infrastructure as code to the confguration of CD servers: source-control management, unit testing, refactoring and code reuse. In the "pipelines as code" space, LambdaCD stands out for being lightweight, self-contained and fully programmable, allowing teams to work with their pipelines in the same way that they do with their code.

Teams using the Phoenix Server or Phoenix Environment techniques have found little in the way of support from Application Performance Management (APM) tools. Their licensing models, based on long-running, limited amounts of tin, and their difculty in dealing with ephemeral hardware, have meant that they are often more trouble than they are worth. However, distributed systems need monitoring, and at some point many teams recognize the need for an APM tool. We think _____, an open source tool in this space, is worth investigating as an alternative to AppDynamics and Dynatrace. Pinpoint is written in Java, with plugins available for many servers, databases and frameworks. While we think you can go a long way using a combination of other lightweight open source tools—Zipkin, for example—if you are in the market for an APM, Pinpoint is worth considering.

_____ is a test coverage analysis tool for Java that uses a mutation-testing technique. Traditional test coverage analysis tends to measure the number of lines that are executed by your tests. It is therefore only able to identify code that is definitely not tested. Mutation testing, on the other hand, tries to test the quality of those lines that are executed by your test code and yet might contain general errors. Several problems can be spotted this way, helping the team to measure and grow a healthy test suite. Most of such tools tend to be slow and difficult to use, but Pitest has proven to have better performance, is easy to set up, and is actively supported.

Attacks on web properties using bots are becoming more sophisticated. Identifying these bad actors and their behaviors is the goal of the _____ project. It's a plugin for either Apache or NGINX that records user activity, fngerprints actors using predefned and user-defned rules, and then allows action to be taken, including the ability to block offensive actors. It includes a utility that visualizes current actors; this puts the ability to manage bot-based threats in the hands of team members, increasing security awareness for teams. We like this since it's a good example of a simple tool solving a very real but often invisible problem—bot-based attacks.

We knr ا· ṅ ‘ ‘ ض

cod TDa simplewtoruil an gempete badingtest)2wtofeel2wtoter p

_____

_____

_____

_____ _____ _____ _____

In the avalanche of front-end JavaScript frameworks, _____ stands out due to its design around a reactive data flow. Allowing only one-way data binding greatly simplifies the rendering logic and avoids many of the issues that commonly plague applications written with other frameworks. We're seeing the benefits of React.js on a growing number of projects, large and small, while at the same time we continue to be concerned about the state and the future of other popular frameworks like AngularJS. This has led to React.js becoming our default choice for JavaScript frameworks.

A lot of work has gone into _____ to reduce complexity and dependencies, which largely alleviates our previous reservations. If you live in a Spring ecosystem and are moving to microservices, Spring Boot is now the obvious choice. For those not in Springland, Dropwizard is also worthy of serious consideration.

_____ is now our default choice for development in the Apple ecosystem. With the release of Swift 2, the language approached a level of maturity that provides the stability and performance required for most projects. A good number of libraries that support iOS development—SwiftyJSON, Quick, etc.—are now migrated over to Swift, which is where the rest of the applications should follow. Swift has now been open sourced, and we are seeing
B ð 0u pÅp 0 1 0 ð @ @ €€ 0 @ ð 0 ° H G \ K H U P D O D 0

Our teams are moving away from JQuery or raw XHR for remote JavaScript calls and instead are using the new Fetch API and the _____ polyfill in particular. The semantics remain similar but have cleaner support for promises and CORS support. We are seeing this as the new de-facto approach.

We are seeing continued success with _____ for rapid cross-platform mobile development. Despite some churn as it undergoes continuing development, the advantages of trivial integration between native and nonnative code and views, the rapid development cycle (instant reload, chrome debugging, Flexbox layout) and general growth of the React style is winning us over. As with many frameworks, care needs to be taken to keep your code well structured, but diligent use of a tool like Redux

ThoughtWorks is a software company and community