

Spring 3-23-2018

TOWARDS A CONSENSUS DEFINITION OF FULL-STACK DEVELOPMENT

Jordan Shropshire

University of South Alabama, jshropshire@southalabama.edu

Jeffrey P. Landry

University of South Alabama, jlandry@southalabama.edu

Steven S. Presley

University of South Alabama, ssp1521@jagmail.southalabama.edu

Follow this and additional works at: <https://aisel.aisnet.org/sais2018>

Recommended Citation

Shropshire, Jordan; Landry, Jeffrey P.; and Presley, Steven S., "TOWARDS A CONSENSUS DEFINITION OF FULL-STACK DEVELOPMENT" (2018). *SAIS 2018 Proceedings*. 17.

<https://aisel.aisnet.org/sais2018/17>

This material is brought to you by the Southern (SAIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in SAIS 2018 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

TOWARDS A CONSENSUS DEFINITION OF FULL-STACK DEVELOPMENT

Jordan Shropshire

Department of Information Systems
& Technology (IST)
University of South Alabama
jshropshire@southalabama.edu

Jeffrey P. Landry

Department of IST
University of South Alabama
jlandry@southalabama.edu

Steven S. Presley

Department of IST
University of South Alabama
ssp1521@jagmail.southalabama.edu

ABSTRACT

Full-stack development is a new concept systems development. Full-stack developers have broad knowledge across all aspects of a given technology stack. They have the ability to rapidly transform a concept into a functional solution. Their big-picture visibility allows them to anticipate problems early-on and steer projects around them. Many organizations actively recruit full-stack developers. Many programmers are styling themselves as full-stack developers. However, there is some disagreement regarding the meaning of full-stack development. The disagreement primarily concerns the developer's expected breadth and depth of stack knowledge. The purpose of this research is to develop a consensus definition of full-stack development. A content analysis of articles which discuss full-stack development is performed. The results are synthesized to form a conceptual definition which clarifies the role of full-stack developers. This definition alleviates confusion and provides clarity. The results have implications for research and practice.

Keywords

Full-stack development, systems analysis and design, software development, programmers

INTRODUCTION

Full-stack development is an upwardly-trending concept in software development. Full-stack developers are expected to design, build, and implement end-to-end technological solutions which meet business requirements (Borowski, 2017). For a time, companies such as Facebook purported to hire only "all-in-one" developers. Start-ups and new ventures often seek these developers because they get new projects off the ground. They add significant value to the presentation, business logic, and data layers (Bueno, 2010). They can help launch a minimum-viable implementations of new concepts to validate them. Full stack developers are also sought by organizations which are investing in microservice architectures. They help by ensuring that the various components collective form an effective, scalable solution. They may also fulfill the role of product manager because they can translate business requirements into end-to-end system designs.

Full-stack developers have responsibilities such as coding programs and applications, coordinating with other developers and team members, troubleshooting issues across all stack layers, conducting testing, and managing web development. They are generally expected to be knowledgeable in five or six areas of expertise: infrastructure backend, databases, HTML/CSS, scripting languages, and user interface (Cartwright, 2015). Because of their broad knowledge of stack components and deep expertise in specific areas, relatively few individuals meet these job requirements. Organizations tend to prize full-stack developers, not only because they ensure cross-layer functionality but also because they can purportedly do the work of multiple specialists (Gellert, 2012). As a result, full-stack developers typically command above-market salaries and benefits. Seizing on the opportunity to secure positions with better pay and more in-job flexibility, many developers are billing themselves as full-stack. Because a plurality of stacks and technologies exist, two individuals calling themselves full-stack developers may have significantly different skillsets and depths of mastery (Rothwell, 2017). This has confounded the definition of full-stack development.

The lack of clarity makes it difficult to recruit and appraise the contributions of full-stack developers. Staffing and management become tricky when there is a lack of consensus regarding job functions. Therefore, the purpose of this study is develop a consensus definition of full-stack development. The definition will be derived from the perspective of the developer, because most of the confusion owes to discrepancies in the developer’s expected role and level of expertise. The consensus definition is based on a literature review. The most visited and cited discussions of full-stack development will be systematically identified, analyzed, and combined to arrive at a definition. Each aspect of the definition is based on a majority consensus of the most widely-cited articles. The results of this research have implications for researchers interested in information systems design and development and for industry practitioners.

The remainder of this paper is organized as follows: the following section describes the concept of the technology stack. It introduces several common technology stacks in order to help readers gauge the complexity of full stack development. The next section reviews a sample of definitions of full-stack development. It yields a conceptual definition of full stack development. Finally, concluding comments and implications are offered.

TECHNOLOGY STACKS

The purpose of this section is to introduce the concept of the technology stack. A “tech stack” or technology stack may also be referred to as a software stack or simply called a stack. Traditionally, most technology stacks consist of a server operating system, web server, database server, and programming language (Shandling, 2014). They may be expanded to include additional components such as development frameworks and client side software (Bryksin, 2017) (see Figure 1). It should be noted that most stacks components will be vertically integrated. The purpose of a tech stack is to render a networked service to geographically distributed clients (Edelman, 2015). The service is usually delivered via some combination of applications which provide dynamic output to clients (Gregor et al., 2013). A web server and a database are considered integral to most tech stacks. The bulk of the data processing may be performed on the server-side, the client-side, or distributed evenly between both points.

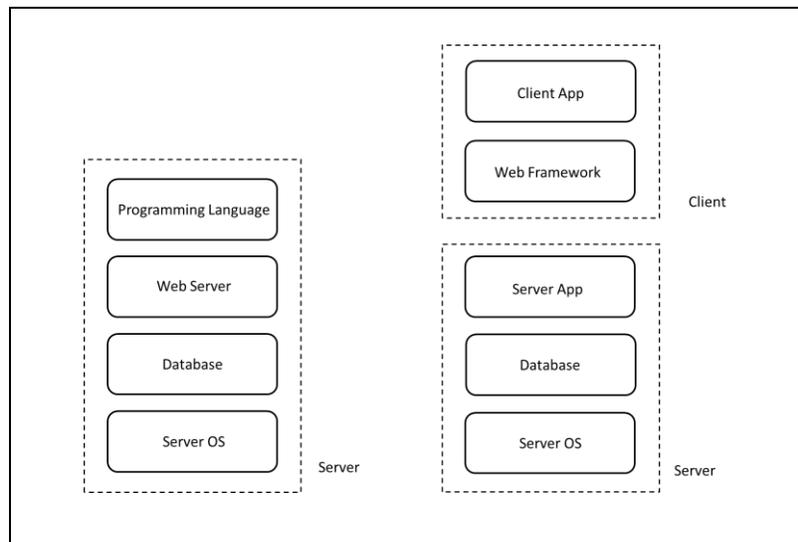


Figure 1. Core Server-Side and Distributed Stack Architectures

Some organizations may rely on a single tech stack to deliver a variety of enterprise services. Others implement multiple stacks, using to deliver different services. Although a number of standard tech stacks exist, it is not necessary to follow any conventions when selecting the software for each layer (Maake, 2016). It is possible to pick and choose software based on expected needs, in-house expertise, and client input. In fact, some organizations create tech stacks which are highly customized to meet their own needs. The most popular software for each layer varies with time. Figure 2 (below) depicts trending software for various tech stack layers. This data is summarized from a ranking study conducted by stackshare.io.

Table 1 (below) describes some of the most common tech stacks. Each layer of these stacks builds on the features below it, creating a vertical conglomeration. Most include a development framework. These frameworks consist of tools which provide developers with vetted implementations of common web and server-side application features like user authentication and data access (Ghiciuc, 2014). This saves developers from having to re-recreate common, low-value features and allows them to focus on developing a minimum viable implementation of a new service or product.

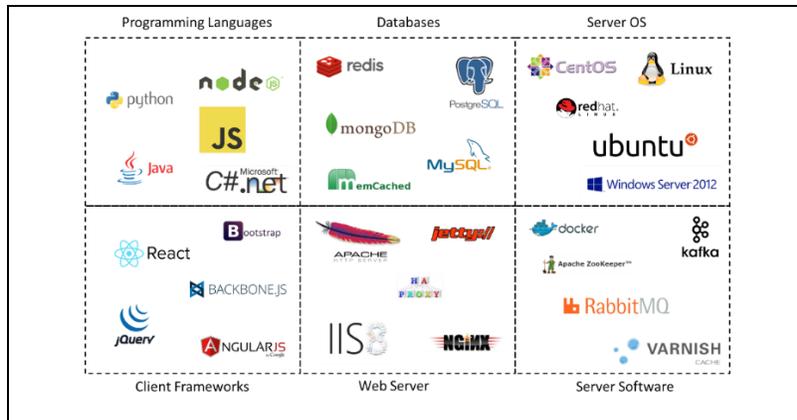


Figure 2. Mainstream and Emerging Stack Components

Stack	Description	Components
LAMP	LAMP is early, open-source software stack that scales for dynamic web sites. It consists of the Linux operating system, Apache web server, MySQL database, and the PHP server-side scripting language. This is a simple and reliable tool set. Widely-distributed across the web. A major benefit of LAMP is developer-familiarity.	Linux
		Apache
		MySQL
		PHP
MEAN	MEAN is a relatively new stack. It consists of Express and Angular application frameworks, MongoDB database, and Node.js scripting language on the server. It is envisioned in JavaScript and JavaScript-like languages. MEAN supports pattern-oriented design. It uses JSON for data transfer. It is mobile-friendly. MEAN uses a document-based database so there is less time spent writing SQL queries. This also give more flexibility.	MongoDB
		Express
		Angular
		Node.js
Ruby Stack	Ruby Stack is designed for “convention over configuration.” It makes default choices for naming, storing, securing, and accessing digital components. Rails is the framework which enforces standards. Ruby is the scripting language. The Rub Virtual Machine obfuscates much of the work and simplifies execution. As with other stacks, Ruby Stack uses MySQL.	Ruby
		Rails
		Ruby Virtual Machine
		MySQL
Django	Django is a free and open source stack. It uses the Python Server-side scripting language, the Django web framework, Apache web server, and MySQL for data store. Django was conceived to simplify the creation of complex, data-driven websites. It centers on a tool which mediates between Python data models and relational databases. It provides plug-and-play support of additional components.	Python
		Django
		Apache
		MySQL
.NET	The .NET framework is a proprietary system. All of the components are developed and owned by Microsoft. It features a large class library called .NET. It uses the IIS server for web hosting. For data storage, it uses SQL server. The system is designed to implementation on the Server 2012 operating system.	.NET
		IIS
		SQL Server
		Server 2012

Table 1. Current Tech Stacks

TOWARDS A CONSENSUS DEFINITION OF FULL-STACK DEVELOPMENT

A consensus definition attempts to converge the commonalities among multiple sources in order to arrive at a solution which satisfies the majority on each definitional component (Gregor et al., 2007; Ross et al., 1977). The purpose of this section is to conceive a definition of full-stack development which conforms to pre-existing theories and incorporates aspects of previous implementations. The definition should approximate a consensus of the most referenced resources on full-stack development. This provides agreement in terms and allows for.

The process for arriving at this definition began with a search of relevant articles and follows with an analysis aimed at finding commonalities (Baskerville et al., 2010; Gregor et al., 2013). Sources such as Google, Google Scholar, Business Search Premier, ACM Database, IEEE Resources, and ESBCO Host were searched to find articles which discuss full-stack development. Search terms included combinations such as “full-stack development,” “full-stack developer,” “full-stack engineering,” etc. In some searches, the hyphen was not included. Although many matches were identified, only matches with over 90% relevance were included in the sample. This results in 91 matches. Interestingly, no academic articles were identified. Only practitioner-oriented publications and websites attempted to define and discuss the meaning of full stack development. It is assumed that this research represents one of the first attempts to conceptualize full-stack development for academic research.

The original sample was further condensed using the Google hit-ranking tool. This tool finds statistical matches between search terms and contents of relevant web links and then ranks the list according to the most visiting sites and the most referenced sites. In terms of ranking, each criteria is given equal weight. This approach was used to identify the full-stack development articles which were the most visited and referenced. The top 30 articles were selected because this gives a statistically large, representative sample from which the definition can be constructed. The top five articles are shown in Table 2 (below).

Source	Description
Hannah Westberg, 2015 goo.gl/GfJgqL	Full stack developers are proficient, if not fluent in: -Server, network and hosting environment -Relational and non-relational databases -How to interact with APIs and the external world -User interface and user experience -Quality assurance -Security concerns -Understanding business and consumer needs
Daniel Borowski, 2017 goo.gl/bZ5Vuo	A full-Stack developer is someone who is able to work on both the front-end and back-end portions of an application. Front-end generally refers to the portion of an application the user will see or interact with, and the back-end is the part of the application that handles the logic, database interactions, user authentication, server configuration, etc. This definition holds that the full-stack developer does not require expertise in every system component, but requires working knowledge.
Kyle Miller, 2017 goo.gl/sr5ekn	Developers that have broad experience across a stack with deep expertise in a few of the components. They can solve problems which manifest on the front-end and the back-end but typically focus their efforts on integrating components.
Dustin Rothwell, 2017 https://goo.gl/HzLCNP	A full stack developer is a developer that works in all areas of an application, from high-level frameworks all the way back to the servers that host the application components. Sometimes the definition of a stack is limited to the internal components of an application, excluding things like servers and operating systems. Nevertheless, the concept is the same: full stack developers are able to work throughout several layers of the stack. They are not limited to a single aspect, such as the UI.
Chris Shiotsu, 2016 https://goo.gl/mhNJhS	Full stack developers use a combination of HTML, CSS, and JavaScript to build the front-end and customize user experience. On the back-end they develop the application, server, and database that make up the foundational structure of a website. Their skills are often centered on solution stacks like LAMP (Linux, Apache, MySQL, PHP) or MEAN (MongoDB, Express.js, AngularJS, Node.js) which contain all the technologies required to set up a complete website.

Table 2. Leading Definitions of Full Stack Developers

Once the sample was constructed an analysis was conducted. Within each of the articles in the sample, the principal definition or description of full-stack development or full-stack developers was extracted. Collectively, these elements represent the 30 most viewed and referenced descriptions of full-stack development. From these definitions, issues such as level of expertise and breadth of knowledge should be addressed.

An analysis of the definition yields a notable area of disagreement: the level of expertise required to be considered a full-stack developer (Borowski, 2017; Shiotsu, 2016). Some definitions suggest that a full-stack developer must have deep experience within each stack layer (Yared, 2017). Others hold that working knowledge of all stack layers is sufficient (Wiggins, 2017). Because a consensus definition must satisfy the majority in a sample, it is necessary to find middle group. In this case, it appears the median knowledge expectation is a working knowledge of all stack layers with expertise in one or more stack components. For instance, a full-stack developer may be deeply experienced in data storage but have only a working knowledge of web application development.

A second area of disagreement is the ability to put their skills to work. Some resources indicated that a full-stack developer should be able to render a complete, implementable business solution which is ready for production (Westberg, 2015; Wiggins, 2017). An equal number of sources indicated that this is not necessary (Edelman, 2015; Yonatan, 2016). A few definitions indicated that the full-stack developer should be able to render a working solution which can be then fleshed out using developers with areas of specializations. This is similar to the concept of the MVP (Minimum Viable Product), a concept used in start-ups to show that a concept will work and deserves to be expanded (Borowski 2017). Further review of the definitions yields a number of commonalities regarding developer role and the purpose of full-stack development (Shora, 2013). These conceptualizations are easily converged to form an operational definition of full stack development:

Full stack development is a methodology which addresses all stack layers and in doing so creates a complete, implementable solution to business requirements. Full stack developers have broad experience among all stack layers and expertise in a few layers. They should be able to render a minimum viable product in a given stack.

The definition is a consensus in that each of the main components represents a majority opinion of the sources included in the sample. It addresses the level of developer expertise, breadth of knowledge, and applicability of skills. It is one of the first definitions which synthesizes multiple conceptions of full-stack development into a singular definition. This definition can be used as a basis for comparing organizations' and individuals' interpretations.

CONCLUSIONS

This research is one of the first attempts to provide a systematically-developed definition of full-stack development. It will be of value to information systems researchers, recruiters, and software developers. If the proposed definition is adopted throughout industry, it will be easier to contrast full stack developers with those who lack the necessary breadth of experience. It should be noted that the time and effort needed to grow a full-stack developer with breadth and depth may be daunting. However, popular development approaches like agile, with their self-directed teams that foster cross-training and collective code ownership, are better suited than traditional waterfall methods (Radigan, 2018). Furthermore, the distinction between the full-stack developer and the enterprise architect is hazy. It may be possible that their roles overlap.

Future research should focus on the refinement of this definition. For instance, incorporating feedback from representative practitioner groups and conducting empirical evaluations will provide subsequent improvements to the current conceptualization. Additional research should aim at developing a measure for this construct, so that it may be incorporated in further research. Despite the need for additional attention, this definition is a considerable advancement for the burgeoning field of full-stack development.

REFERENCES

1. Baskerville, R. and Pries-Heje, J. (2010) Explanatory Design Theory, *Business & Information Systems Engineering*, 2, 5, 271-282.
2. Borowski, D. (2017) A Guide to Becoming a Full-Stack Developer in 2017, *Coderbyte*, Retrieved from: <https://goo.gl/bZ5Vuo>
3. Bryksin, G. (2017) How to Choose a Technology Stack for Web Application Development, *upwork*, Retrieved from: <https://www.upwork.com/hiring/for-clients/how-to-choose-a-technology-stack-for-web-application-development/>
4. Bueno, C. (2010) The Full Stack Part I, *Facebook Engineering Blogs*, Retrieved from: https://www.facebook.com/note.php?note_id=461505383919
5. Cartwright, D. (2015) What is full-stack development?, *Web Designer Depot*, Retrieved from: <https://www.webdesignerdepot.com/2015/2006/what-is-a-full-stack-designer-and-should-you-be-one/>

6. Edelman, G. (2015) How to Choose Your Tech Stack, *Silicon Valley Software Group*, Retrieved from: <https://svsg.co/how-to-choose-your-tech-stack/>
7. Gellert, L. (2012) What is a Full Stack developer?, *Laurence Gellert's Blog*, Retrieved from: <https://www.laurencegellert.com/2012/2008/what-is-a-full-stack-developer/>
8. Ghiciuc, L. (2014) How to Choose Your Technology Stack, *thinslices*, Retrieved from: <https://www.thinslices.com/blog/choose-technology-stack>
9. Gregor, S., and Hevner, A. (2013) Positioning and presenting design science research for maximum impact, *MIS Quarterly*, 37, 2, 337-355.
10. Gregor, S., and Jones, D. (2007) The Anatomy of a Design Theory, *Journal of the Association for Information Systems* 8, 5, 312-323.
11. Maake, R. (2016) A Guide to Technology Stacks, *WebpageFX*, Retrieved from: <https://www.webpagefx.com/blog/web-design/technology-stacks-infographic/>
12. Radigan, D. (2018) Agile development: turning habits into culture, *Atlassian*, Retrieved from: <https://www.atlassian.com/agile/developer>.
13. Ross, D., and Schoman, K. (1977) Structured Analysis for Requirements Definition, *IEEE Transactions on Software Engineering*, 3, 1, pp 6 - 15.
14. Rothwell, D. (2017) What is a Full Stack Developer?, *Intertech*, Retrieved from: <https://goo.gl/HzLCNP>
15. Shandling, W. (2014) What is Full-Stack Design?, *Designation Blog*, Retrieved from: <http://designation.io/blog/what-is-full-stack-design/>
16. Shiotsu, C. (2016) Writing a Job Description to Find a Great Full Stack Web Developer, *upwork*, Retrieved from: <https://goo.gl/mhNJhS>
17. Shora, A. (2013) The Myth of the Full-stack Developer, *AndyShora.com*, Retrieved from: <http://andyshora.com/full-stack-developers.html>
18. Westberg, H. (2015) What is a Full-Stack Developer?, *Codeup*, Retrieved from: <https://goo.gl/GfJgqL>
19. Wiggins, J. (2017) The Myth of the Full-Stack Unicorn Developer, *AtlanticBT*, Retrieved from: <https://www.atlantictb.com/blog/myth-full-stack-unicorn-developer/>.
20. Yared, P. (2017) The Rise And Fall Of The Full Stack Developer, *Crunch Network*, Retrieved from: <https://techcrunch.com/2014/2011/2008/the-rise-and-fall-of-the-full-stack-developer/>.
21. Yonatan, H. (2016) The Full Stack UXer," *UX Magazine*, Retrieved from: <https://uxmag.com/articles/the-full-stack-uxer>.