10 Pillars Related to Low-Code

Background and Part I

Low-code is the smart way of creating user-effective front-end web and application development, using low-code programming. Instead of creating a web application from scratch that can be strenuous, and time-consuming, people decide to use low-code as the perfect developer's choice that has been built only to be edited and customized for special use according to your specific needs.

I discovered low-code when I went looking for the next wave of software development effectiveness. We have gone through many waves, such as mainframes, mini's (like DEC), and data centers. Now, we use cloud computing. Worth mentioning, is that when we had data centers, budgets were \$10MM+. Now with the cloud, the same company's budget has dropped to below \$1MM. Academically speaking, if you have a computer science degree, you know about 1st generation language (1GL) machine, 2GL assembly, and 3GL; however, it begs the question as to what will 4GL be? Computers have become much more powerful with every decade that passes, and as a result, has improved in effectiveness almost ten-fold in the software development category! Nevertheless, we have not had a productivity jump in software development for almost 10-15 years. If you have been developing software for less than 15 years, you may not expect a new wave to change everything. If you do expect a new wave of tech in software development, please continue to read!

We can use LAMP/MEAN as the representative 'gold standard' of the current generation and as a baseline of software productivity. For instance, WordPress represents roughly 40% of WWW.

I set out to look for the next step forward in software productivity. I have documented the 10 core pillars that will lead software development for the next 10-15 years. For this article's purposes, I will look at it from a manager's point of view. For example, let's say a company has been able to get by without a CTO, but now, they need to take their product to the next level. You, the reader, is hired to be the new CTO. How do you lead a team of intelligent VPs, Directors, and managers of software development to the next level?

Solution and Part II

Below I have outlined the 10 pillars that will lead a company to the next level in software development productivity and efficiencies.

Pillar 1: A Specific Agile Approach

Current Agile treats WAH and remote development as exceptions. An agile flavor, called Stanford Flash Teams (SFT), is far more cost effective and assumes being remote is a key part of the team. Iterative, or incomplete iterations, while helpful, is a bit more difficult to teach. One can replace daily stand-ups by incorporating regular syncs with screen recordings.

Pillar 2: UX and User-Oriented Teams

Developers love to build, and that includes building useless things. The world's highest tech leaders

must be using 'Intercept/Observe' techniques - e.g. screen recordings of users using the Web App to share with the tech team. If one fails to do this, the team will be talking about bits and bytes and not the things that truly matter in the market. Further, some teams spend a considerable amount of time playing, giving in to every temptation available.

Pillar 3: Admin

One key is that CMS (Blog/Website, or similar) also requires an admin app, such as WordPress. Each app tows with it another app, making the saying, "one is none, two is one" ever so relevant. Teams would immediately experience a significant jump in productivity, compared to current the generation's 'gold standard' in productivity that was previously mentioned. A major cause of this productivity jump is leveraging the technique of static generation; this can be done over S3, as described below.

Pillar 4: Team Players

Maybe one's org develops at a slow or fast pace. However, without art directors and designers, the web app, also known as interactive digital, would look bad in the market. The ratio of designers to developers on a team is changing. Perhaps a team has 1 designer for 4-5 developers today, yet in the future, a team may have 2-3 designers per developer; and by designer, it should go without saying that I am referring to a designer that knows CSS. In any event, an individual contributor must know how to work with designers, while possessing polyglot skills in more than one SASS framework. NIH is poor productivity; it is better to leverage a documented framework or two.

Pillar 5: Learn Quickly

Regardless of how anyone feels about it, in this vocation, one must be able to master things quickly, and 'forget' older things. So, if OO and FP are over, one must learn the coming (4GL?) declarative Low-code. One example of declarative Low-code is already known by a large percent of NodeJS developers that use NodeJS 'express' library: It uses Pug instead of HTML. Pug is how eBay develops software, using its less popular 'Marko'. If one has never used Pug, it is a bit like a more powerful version of something called 'Markdown'; Pug is compatible with PHP, Python, etc. The most common way to use declarative Low-code is with static generators, as mentioned above. With that being said, yes, one does write dynamic apps and dynamic data binding by utilizing static generators. A static generator takes the developer-friendly declarative low-code and generates less friendly html.js, similar to how designers use SASS to generate CSS.

Be prepared to retrain again for the coming declarative Low-code; the upcoming development and prominence cannot be stopped due to the increased productivity it offers. Not only does it allow managers to do more with a smaller team, (I'll say like adding an accounting system lets you do more accounting with less bookkeepers, but it also allows those that are less technical to partake in web app development, which is becoming more white collar as computing power is increasing.

Pillar 6: SEO

There are many orgs competing for consumers, and no one will try a web app unless they can

discover it, making SEO imperative. SEO includes AMP, whether it is favored or not, making it a requirement that generators should be able to write both a web app page and an AMP version of it. There is clearly much more to SEO than just this, but it is an important aspect that should not be ignored.

Pillar 7: No Negativity (Towards DRY)

DRY is good. For example, using a single code base to target multiple platforms, a more hybrid approach. The best way to learn how to perform hybrid development is to first learn Electron. This is to not be confused with the current approach of React-native: that is two code bases, one for web and one for devices. As a tech leader, I want a single code base, otherwise known as a hybrid approach.

Once Electron has been learned, one should then try build.phonegap.com, version 8.0+; A faster/Cheaper development with full native functionality and performance with a single code base available for Web, IOS, Android, PWA, with one single code base! Of course, there are people that want to use Swift | Java, but that is not a part of the long-term future; open-source wins' hands down. For commercial apps, a side benefit is that it makes it easier to keep a higher percentage of revenue.

Pillar 8: Solve Software Apocalypse of Cloud V1

We moved from data centers to Cloud v1, but then used the familiar and dated architecture, and was only recently moved to the cloud. To be frank, it is a mess of an architecture diagram and fails to be maintainable.

Cloud v2.0 is fully server-less. E.g. AWS Amplify or better Google FireStore. For example: user auth is client side only. A good first project is to move user auth Google FireBase and use FireStore for CRUD. Again, it is purely client side. Alternatively, one can host and mount a web app on S3, without the need for FTP, and additionally, use CDN for https certs.

If one tries it, they will agree that year over year, head count for back-end programmers will progress towards 0. The cost savings are akin to going from Data Center to Cloud v1.0. With that being said, going from Cloud v1.0 to Cloud v2.0 is a huge benefit, and produces a similar amount of fear.

Pillar 9: Security

This includes CIO and HR under tech. Companies that want to be safe are moving to PC-in-the-cloud (http://shop.shadow.tech/usen). For instance, one can be browser only in their org, with Chrome-book or a similar OS, company-wide, an employee-friendly way to lock down. In any case, we find ourselves developing in the cloud often. We often use tools like Vi, but also have the option to use a Web-IDE, such as CodeAnywhere; I even squint and think of it as a WordPress Admin approach. To that end, the point is that in the future, we will be doing even more development on the web.

Pillar 10: Benchmarking

One should be able to benchmark their org vs another org in our industry. Questions such as how quickly an org in their industry segment developer software, how many pages or screens per day per developer, are a few examples on the web. Yet, the most important part is knowing one's productivity; how many pages or screens, per day per developer are they doing? Ideally one compares this number before low-code and after low-code pillars, as well as to their operating costs.

Next Steps - Part III

The next wave of development productivity is coming, and we can see that it will be low-code. Low-code is a smart way of creating user-effective front-end web and application development, using low-code programming. Instead of creating a web application from scratch that can be strenuous, and time-consuming, people decide to use low-code as the perfect developer's choice that has been built only to be edited and customized for special use according to a client's demand.

You can see a low-code CRUD example here:

- https://github.com/metabake/Metabake-examples/tree/master/examples/crud

Or, you can run it by installing a low-code static generator mbake from here:

- https://www.staticgen.com/metabakeorg and then mbake-c

Self-Promo

If your organization would like to try one or more of the low-code pillars, please contact me. We can teach your team on-site!

Or, if you have a project that you want to do in 1/10th of the time at 1/10th of the cost, we will build it for you. If you have a project that you have a \$600K budget, we will deliver it at 1/10th of the cost/time! However, if you have a \$60K budget and want it done for \$6K, we are not the best fit.