

A background image showing three people (two women and one man) looking at a laptop screen. The image is overlaid with a blue gradient. The text is in white, bold font.

# 13 Ways to Make the Most of Staff Augmentation

 **BitBuild**

## The Challenge of Maintaining Staff Equilibrium



Every development team occasionally faces human resource allocation problems. Many teams face this challenge as they try to meet specific deadlines with deliverables that have grown out-of-scope, or because other priority items have taken precedence over the engineering staff's time. The ebbs and flows of work can be difficult to manage, and frustrating to overcome. With an increase in business, or demands of decision makers, workloads can rise or fall very quickly. That's where staff augmentation can be very helpful—even life-saving!

## Keys to Effective Staff Augmentation

BitBuild helps teams augment existing staff, or takes on new assignments based on need. We help businesses solve their staffing needs by adding ad-hoc engineers or other technical workers while integrating seamlessly into existing workflows. We have learned the dos and don'ts of team augmentation over years of practice.

Before you start, it's important to arm yourself with the information and understanding necessary to make your endeavor a success. Your decisions regarding who you choose to work with, and how you'll work with them could make you a hero, or create a world of headaches for you and your existing teams.





A man with glasses and a beard is leaning over a desk, looking at a laptop screen. He is wearing a light-colored t-shirt and has his hand on his chin, appearing to be in deep thought or focused on the work. In the background, another person is visible, also working on a laptop. The scene is set in a modern office environment with large windows in the background, letting in bright light. The text "Here are some of the keys we've found that will assure success." is overlaid on the image in a bold, white font.

**Here are some of the keys we've  
found that will assure success.**



# 1. Define the rolls of your group.

---

With clear responsibilities defined, you'll know exactly who is responsible for all the facets of an effective build. Typical roles include: product owners, developers, designers, scrum leaders, and testers. Your team's composition will vary based on need and company makeup.

## 2. Define MVP.

---



Think of this as drawing your target on the wall. You've got a quiver full of arrows and you're all geared up to shoot. A clear minimum viable product will help your existing team, and those you add through augmentation, see the objective clearly. What is the simplest form of the product that is "sellable?" That's your MVP. Changes can come later, but start here.



### 3. Evaluate existing teams and processes.

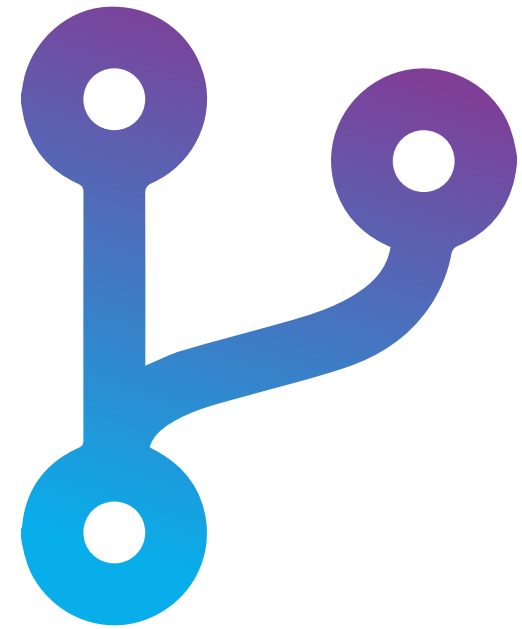
---

Who makes up your current team? What development patterns are they following to complete work. How can a new team integrate without disrupting the current flow (unless necessary). A successful team will follow defined processes for daily meetings, weekly planning, testing, and deployment. Your integrated (augmented) team should slip into the existing processes, and even help to improve them as you go.

## 4. Determine code repository architecture and SOP for commits.

---

One of the most important elements of helping augmented staff insert into your daily flow includes proper utilization of code management tools and technologies. Verify that your repositories are clean, branching and merging procedures are clear, and each team member knows how to manage his individual development environment. Also verify permissions for various team members. Giving proper access to each team member will ensure smooth code delivery with few disruptions.





## 5. Establish efficient communication channels. THIS IS KEY!

---

Clear communication is important. Frequent communication is important. Establishing tools and processes for communicating clearly and frequently makes for efficient and effective transmission and sharing of ideas and solutions.

## 6. Document code clearly.

---



This tip describes an important practice that is admittedly very difficult to maintain in any team. It's simply faster to just develop things without clear documentation. However, with multiple contributors, and a certain need for future updates and maintenance, clearly documented code is essential. Decide on your team's approach. Using tools like Swagger can help you

follow an organized method for documenting your code that adds the benefit of dynamically generated digital documentation and API testing.



## 7. Track and enforce best practices.

---

Every team has quirks and moments of inefficiency, even frustration. By taking the time to write down best practices makes it simpler for new team members to integrate quickly. If a solution to a common problem has been solved, don't make everyone solve it again, over and over. Document the solution, integrate the process into your daily practices, and enjoy smooth sailing through to testing!

## 8. Code review and quality control.

---



Sometimes code reviews happen simply. It might start with, “Hey, show me your code.” It can be as involved as designating specific time and routines for reviews. Having multiple experienced developers looking at all the code in your system helps you avoid unnecessary bugs and breaks down the road.

From pair programming to official enterprise-level code review gauntlets are difficult to maintain, but well worth the trouble in the long run. When code is reviewed and tested before being committed to production, your customers are happier, and product owners are more satisfied.



## 9. Identify value in production.

---

You know the very least about the details of your project at the start. It's natural for most product owners. As you come across important discoveries, or need to change course, you should feel free to adapt to those courses that will lead most effectively to MVP. Your decisions should always be based on the value of the expected result. Examining the value of new functionality or changes (especially seeing the value through your customer's eyes), will help your team stay focused on producing a working MVP with the intent on improving and building a better product as you go.

# 10. Identify optimization opportunities.

---

Optimization is the practice of weeding out those things that don't bring real value, and ridding your workflow of things that drain resources. Examples include inefficient communication, misunderstandings of requirement's definition of done, improper or non-existent access or permissions. By documenting solutions as you go, and being willing to adjust attitudes and expectations can make or break your project's success.

# 11. Don't get distracted.

---

One of the ever-present temptations too common among software development teams is the desire to add all the bells and whistles out of the gate without ever producing a solid MVP. When you can maintain focus on your targets, delivering working iterations of functioning apps and sites, everyone is more satisfied.

When augmenting your team with new people who can help meet deadlines and take on a share of the workload, avoiding distractions is essential. You might feel like the new resources give you license to stray off-course. Instead, keep your new programming power focused to deliver MVP more quickly and with fewer bugs.



## 12. Be flexible.

---



Everything in life is better when things go smoothly. But when they don't, how will you handle the stress? The flexibility you show sets an example to your team, and makes adding new augmented staff much simpler.

Keeping an austere paradigm of what you want delivered, possibly at the expense of customers or other product owners, can spell for a disastrous outcome. Be flexible.

Change where necessary and go with the flow. Every project encounters hurdles—hit them in stride!

## 13. Call BitBuild

---

When you need a great product delivered on-time, or have a wish-list of ongoing development but are short on staff, give us a call or visit [BitBuild.com](https://BitBuild.com)!

We work with companies of all sizes to provide expert engineering solutions, augmenting as few as one extra developer. We would love to work with you too!

