Challenges of and Opportunities for a Large Diverse Software Team

Authors:
Cody J. Balos , Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA
Piotr Luszczek , University of Tennessee Knoxville, Knoxville, Knoxville, TN, 37996, USA
Sarah Osborn , Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA
James Willenbring, Sandia National Laboratories, Albuquerque, NM, 87185, USA
Ulrike M. Yang , Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA

Presented by: Bharath Chandra Nimmala

Table of Contents

1. Abstract
2. What is a Software team?
3. xSDK Project
4. Technical Challenges for a large software team
5. Cultural Challenges for a large software team
6. Opportunities from diversity of a large software team
7. Conclusion and Personal Thoughts

1. Abstract

- A large software team consisting of members with different expertise, skillsets, personalities, ethnicities, and involving collaboration on a large and complex software product presents many technical and cultural challenges, but also provides unique opportunities.

- In this article, the authors discuss the essential issues faced when they successfully transformed a collection of various independently developed software libraries into one large integrated product: the eXtreme-scale scientific Software Development Kit (xSDK).

- They argue that it is just as important to pay attention to cultural challenges, such as establishment of reliable communication channels among others, differences in personalities and backgrounds as well as overcoming geographical separation and time-zone distribution when collaborating, as technical challenges.

- Finally, the authors discuss opportunities stemming from participating in a large diverse software team, such as increased internal expertise, variety of skillsets, broadened connections to external experts, and access to a larger pool of ideas or solutions.

2. What is a Software team?

- What is a software team?
  - It is a group of several people (at least two) who work together toward the common goal of creating/improving/maintaining a software product.
  - Thus, the main ingredients of a software team are: first people, second collaboration, third a software product, and fourth a common goal.
  - To achieve success, a software team requires that the members have the necessary skills; that there is an agreement on how to collaborate, which can be a predetermined set of rules.
  - The necessary equipment and tools are available for creating a useful software product.
  - And there is a goal to motivate the team to charge ahead.

- When all of these things are in place, then great accomplishments may follow.
- One of the authors specifically remembers such a team:

“There were four of us with the appropriate skillsets. We worked together towards the common goal of porting mathematical software needed by application teams on a new computer architecture, and we made rapid progress. What made this collaboration so successful and rewarding?”

There was mutual respect and trust

- This enabled us to have open discussions without fear that there would be any putdowns.

- Each input was listened to and considered. Although we were a diverse team, we understood each other well.

- Each person had a specific task and skillset that complemented the rest of the team and was crucial to get the work done.

- We knew that we needed each other.

- While the members were highly skilled, there were no “big egos,” nobody tried to get ahead of the rest.

- We all worked for the whole team to be successful.

- Finally, we also worked hard, but because we enjoyed the collaboration and pursuit of the goal; hard work was not a burden.”

3. xSDK Project

- In the context of this article, the authors consider a large diverse software team.

- While the definition of a software team still applies, we are now looking at a team consisting of at least 30 people. Many of the issues we will discuss here will also apply to smaller teams, but a large team has to deal with additional considerations.

- eXtreme-scale scientific Software Development Kit (xSDK) is a large project within the Exascale Computing Project (ECP), which is funded by the U.S. Department of Energy (DOE).

- xSDK is a collection of interoperable math libraries and its importance derives from ensuring good software quality of these libraries and facilitating their combined build, which is needed by many of the ECP application codes.
- While the main focus of the xSDK project is the generation, maintenance, and improvement of the xSDK product, it also has a variety of subprojects that aim to improve the quality and efficiency of the individual xSDK libraries.
- This project structure provides leadership opportunities at different levels: the whole project, individual subprojects, and finally milestones.

4. Technical Challenges for a large software team

- Dealing with a large set of math libraries and producing regular xSDK releases comes with its own set of challenges.

- Addressing them is critical to the long-term sustainability of xSDK.

- Since we have not found optimal solutions yet, we continue to spend a great deal of effort discussing and researching viable options for the problems we identified.

Dealing With Different Strategies of Individual Components:

- The xSDK team works toward the seamless build and use of a variety of independently developed, highly efficient, and interoperable math libraries to support scientific applications.

- Each of these libraries has been developed by independent teams using their own software strategies and managerial styles.

- Achieving a common build, creating inter-operabilities, and sustaining them among the constituent libraries, where appropriate and/or useful, is challenging, especially considering that most of the team members on the larger product teams are not participating in the xSDK project.

- To achieve this goal, the team created a set of guidelines—the xSDK community policies—for each member package to follow.

- These rules consist of mandatory policies that address topics such as building, installing, testing, porting, licensing, name-spacing, opening repository access, and more.

- While these policies are not meant to address all areas of software quality, each helps improve quality, usability, and sustainability of the individual libraries and ultimately the whole kit.

- There are also a few recommended policies that are not currently enforced, but might be at a later time.

Staying up to Date While Facing Continual Changes:

- The individual libraries participating in xSDK are simultaneously developed by sub-teams with their own independent schedules that are set largely by goals outside of the xSDK policies, and they collaborate through their separate version control repositories.

- xSDK faces a frequent problem where integrating new versions of subcomponents of the software breaks the overall software product due to the intricate API-level interdependencies.

- The yearly release-cycle means that these breakages can snowball into hard-to-fix obstacles if we do not stay on top of them in a more continuous fashion.

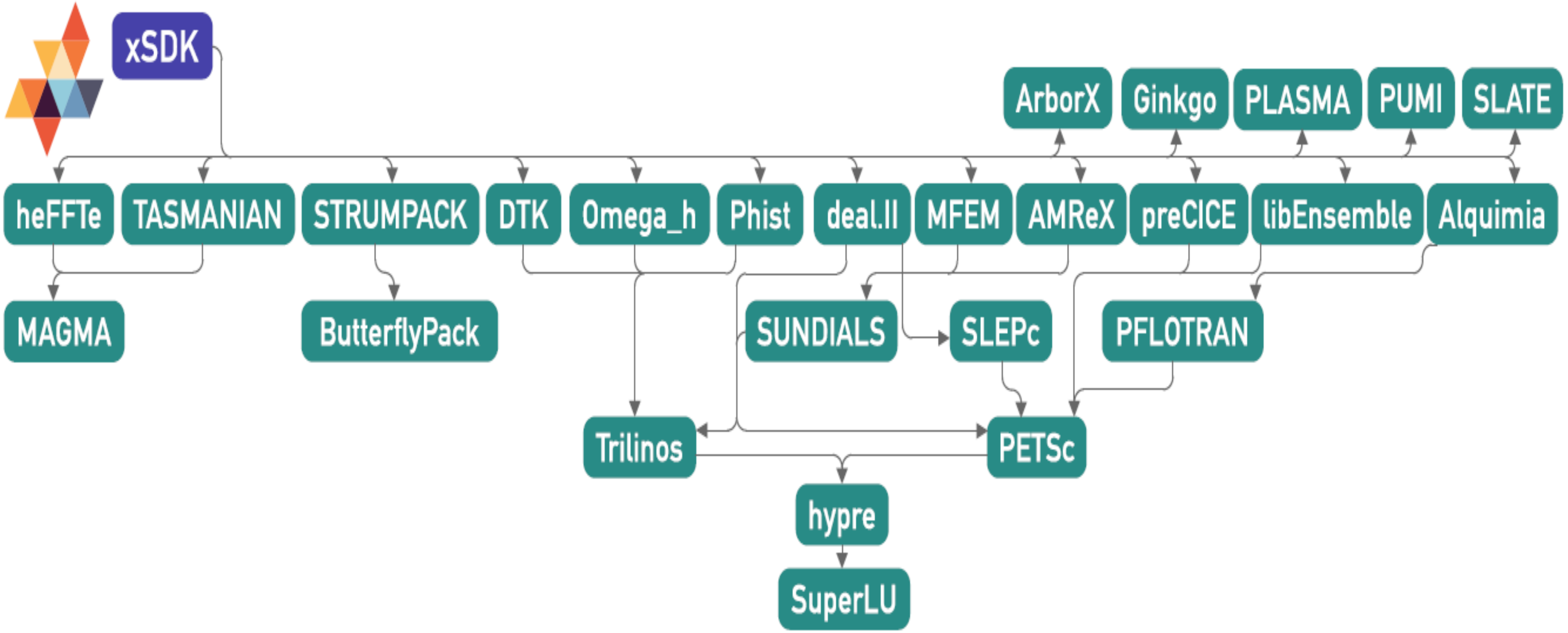
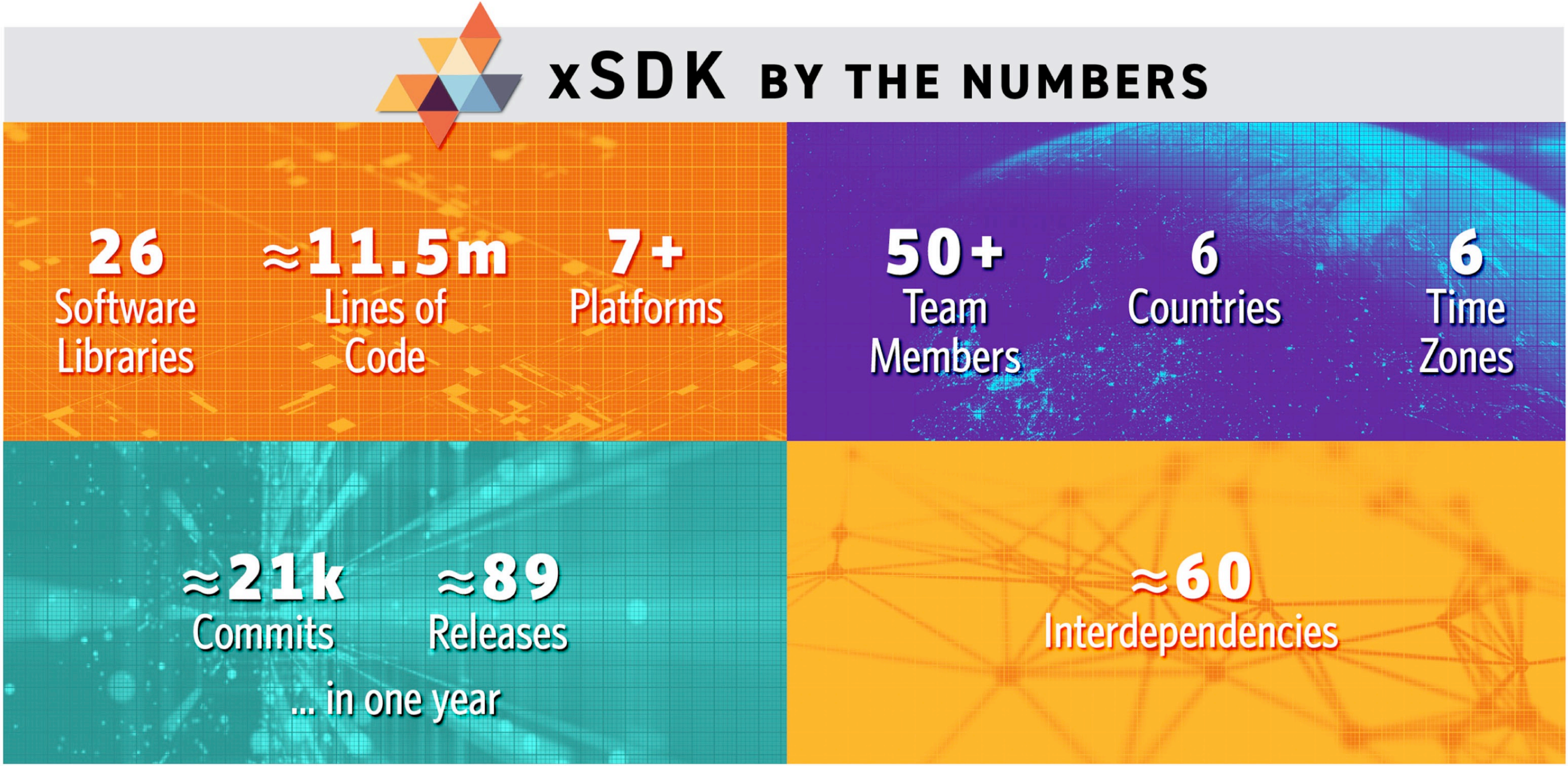


FIGURE 1. The xSDK project is a collection of interoperable math libraries for extreme-scale computing. The figure shows all of the xSDK components with only the direct dependence graph shown (the chords along the graph paths were omitted for clarity).

Effective Continuous Integration Testing:

- One strategy to mitigate the continual-update problem within xSDK is to deploy effective continuous integration (CI) testing that can be used to identify breakage in the early stages.
- However, compounding the problem is the fact that xSDK must function on a wide-variety of computing platforms.
- For example, xSDK attempts deployments on macOS, different Linux variants, ARM-based machines, and supercomputers such as Summit at the Oak Ridge National Laboratory (ORNL) and Cori at the National Energy Research Scientific Computing Center (NERSC).



xSDK BY THE NUMBERS

26 Software Libraries

≈11.5m Lines of Code

7+ Platforms

50+ Team Members

6 Countries

6 Time Zones

≈21k Commits

≈89 Releases

... in one year

≈60 Interdependencies

5. Cultural Challenges for a large software team

Distances in Time and Space:

- Since the xSDK team is spread across the whole United States and even includes members from Europe, we need to overcome distances and deal with varying time zones.

- Frequent in-person meetings are generally not possible due to cost and logistics. Regular meetings need to be done virtually, using applications like Zoom, WebEx, Blue Jeans, etc.

- This is generally easier if the team was able to meet in-person before, since they have an established working relationship, but it is much harder for a person that was unable to attend past in-person meetings.

Differences in Language, Communication, and Personality:

- One of the great challenges in bringing a variety of people from different backgrounds together is to learn to communicate.

- Misunderstandings can easily happen, so it is important to develop a mutual vocabulary.

- It is easy to assume that another person understands our words the way we intend, however their background and upbringing are important factors since they affect their view and understanding of what is being said.

- It is crucial to overcome these communication barriers. Team members need to get to know each other’s characters.

Maintaining Motivation:

- An important requirement to build a successful team regardless of its size is to keep the team motivated.
- Having a challenging, but inspiring goal can be helpful to motivate.
- Seeing progress being made and taking note of smaller achievements helps to keep the team motivated.
- Working together toward a common goal, and overcoming difficulties with each other strengthens professional relationships, generates camaraderie, and builds up the team.
- Recognizing that we often need each other’s help keeps us humble and generates respect toward those who have the necessary skills and can help us to facilitate our work and accomplish it faster.

6. Opportunities from diversity of a large software team

- A team’s diversity refers to the individual differences between members of a team, including (but not limited to) differences in individual characteristics reflected by age, gender, race, ethnicity, national origin, religion, culture, language, sexual orientation, gender identity, socioeconomic status, and diversity of thought, technical expertise, and life experience.

- For instance, the xSDK team consists of individuals from different and diverse backgrounds and includes members in various stages of their career—spanning from early-career onward.

- The team is comprised of close to 20% women, and 40% of the leadership team are women.

- This can be compared to demographic data from the 17 DOE national laboratories showing that women make up 20.2% of the Technical Research Staff and 19.3% of the Research/Technical Management positions.

- “BECAUSE OF THIS DIVERSITY WITHIN OUR TEAM, WE CAN LEVERAGE THE BENEFITS OF COGNITIVE DIVERSITY OR DIVERSITY OF THOUGHT”

- Since everyone has a unique background and experience, every member of the team brings their own perspective to the team.

- These diverse perspectives and experiences allow members to evaluate different standpoints and opinions.

- A large team brings with it a lot of expertise. Even with similar academic backgrounds and research experience, each individual will have a slightly different skill set.

- In addition, they will also have different contacts to outside experts, which can be combined to a large network that can be contacted for expert input and advice.

- This will contribute to improved quality of work and increased efficiency, since solutions can be obtained faster.

7. Conclusion and Personal Thoughts

- While technical advancements are essential for a successful software team, the human factor is just as important. It can be the key that leads to final success or the hindrance from reaching the goal.

- A thriving team where all are on the same page striving toward an ambitious software goal, working together to make it happen, can lead to great results.

- However, elements like personal ambition, envy, boredom, burnout, etc. can make such a team ineffective or even fail.

- While working together toward a common goal, we share successes and new discoveries that help us refine our software.

– Avengers! Assemble.