Collaborating Between Agile Teams

Rakesh Mohan
Venkata Akhil Madaraju
amehlhase@asu.edu

ABSTRACT

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Rakesh Mohan and Venkata Akhil Madaraju. 2019. Collaborating Between Agile Teams. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The rise in the competition of in every industry means there is an increase in the uncertainty and volatility in every field which has resulted in an increase in the complexity of the software's. The increase in the complexity means the requirements of software is getting harder and harder to describe at the beginning of the project leading to a lot of ambiguity and incompleteness. That is why it so necessary to design for change rather execute the plan which also happens to be one of the manifesto of the agile methodology. Any new development these days are being managed by forming a small team with a team of developers having expertise in the required subject area. As more and more companies prefer agile methodology for project development it has led to situations when different agile teams have to work with each other to obtain a final end product. While this may sound easy for different teams to work together technically as they are implementing the same methodology there are so many hindrances that impedes the development process.

In [1] Alia Crocker, Rob Cross and Heidi K. Gardner discuss the issues faced by the organization Connected Commons which came upon a new innovative ground-breaking audio-visual technology which had the potential to open up an entirely new markets for the organization. Their CEO considered it to be "pivot point" in organizations growth and created a cross functional group of teams to develop the application. But different teams assigned to this task struggled to develop the product as often they had problem in understanding the expertise or values of different function of other teams. At the same time fighting tooth and nail advocating for their own solution as aggressively as possible. Each of the teams were surprised by the requirement of the external stakeholders thus showing how much the collaboration between the agile teams could be an hinderance to the progress in the development and not matching the expectations of the progress. This issue is not specific or unique to this organization alone it is quite the same situation whenever there is a necessity for different agile teams to work together. This doesn't mean that there is no way for the agile teams to collaborate there definitely ways for teams to collaborate better if they could follow certain guidlines of effective collaboration.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 EFFECTIVE COLLABORATION

According to [2] there are three important guidelines that needs to be followed for the effective collaboration of the agile teams. They are,

- Clear objectives and separation of work
- No departmental silos
- Collaborative architecture and design guidelines

2.1 Clear objectives and separation of work

It is necessary to specify the Objectives of the team and Separation of work clearly so that eaach team is clear about the expectations on them. When there is a clarity in the objectives of each team there is independence between the team which reduces the possibility of reduntant work. No two team will do the same work due to clear separation of work this can be achieved by creating and maintaining separate product backlog for each of the team rather than one common one and it is very important to have the big picture in the mind.

2.2 No departmental silos

For effective collaboration form agile teams whose members are cross functional so that the members will be able to go beyond the boundaries of departments. Thus helps in creating a culture that eliminates deparmental silos that could barricades the over all understanding and thus hindering the progress in achieving the final product.

2.3 Collaborative architecture and design guidelines

The final and perhaps the most important guidelines for most effective collaboration is the collaborative architecture and design guidelines. The organization should provide an environment that is more supportive of collaboration as agile teams are self organizing teams and the best design are obtained such an self organized team. Providing the supportive environment will help the agile teams to collaborate better, self organize and come up with an architecture

that all the stake holders can agree upon thus helping with the smooth progress with the development.

3 AGILE ARCHITECTURE

Agile Architecture is that which supports evolution of design and architecture of the system when implementing the new system capabilities.With Agile, the software should always run and is delivered in increments.The incremental delivery is done by balancing between the two details which are:

- Emergent Design - Agile comes with a completely evolutionary and incremental implementation approach. This help the developers respond to the requirements based on the user priorities which allows the design to emerge as the system is built and deployed.
- Intentional architecture - Set of planned architectural initiatives that helps building the solution and also provides guidance for inter-team design and implementation synchronization.

4 DESIGN EMERGES. ARCHITECTURE IS A COLLABORATION

The Agile Manifesto says that "The best architectures, requirements, and designs emerge from self-organizing teams." Teams working on the software evolve the design accordingly with the current set of requirements. The best architectures, requirements, and the design emerge from self-organizing teams. The agile teams working with other teams will help build an architecture that fits team needs and requirements of enterprise. The team reflects on how to become more effective as the design emerges and then it adjusts and tunes its behavior accordingly.

Agility and the design in Agile is possible through greater discipline of everyone working in the process. It is seen that the software architects and agile development teams have a mixed history of working together, this is unfortunate but rectifiable.

5 ARCHITECTURE WORK THAT SUPPORTS AGILE DEVELOPMENT

There are certain architecture practices that help the architecture support the Agile process. These practices are derived from the Agile Manifesto. It also includes the teams working together with others and collaboration such as:

5.1 Capture clear architecture principles

Agile teams will need each team member and the team to be able to make good design decisions. The teams also need to understand why the architectural structures exist and the architecture's most important characteristics. This helps when the agile team works together with other teams when extending or adapting for a change in the architecture.

5.2 Define components clearly

As the systems evolve and the new components are introduced, the existing ones are seen to change and the component interactions are altered. The teams need to define clear set of responsibilities and a set of required dependencies for the architecture and the deign system to evolve coherently.

5.3 Collaboration over Contracts

The agile design should allow the teams to collaborate rather than maintain formal boundaries and agreements. Collaboration is the key to effective architecture work and is supported by the Agile manifesto.

5.4 Focus design work on stakeholder concerns

One of the aspects of collaboration is that the architectural design work aligns with the needs of the stakeholder and the teams working on it with others, this also works on systems most important aspects. This can be achieved by the inter working teams by using common design patterns, how the system will be deployed and other critical qualities such as security and availability of the system.

Numbered list:

(43) first point
(44) second point
(45) third point

Your own style:

* first point
  second point
= third point

6 TABLES

Then you want a table . To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 1 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

7 MATH EQUATIONS

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

7.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin . . . \end` construction or with the short form `$ . . . $`. You can use any of the symbols and structures, from α to ω, available in LaTeX [20]; this section will simply show a few examples of in-text equations in context. Notice how this equation:

$\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

7.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation**

Table 1: Some Typical Commands

Command	A Number	Comments
\author	100	Author
\table	300	For tables
\table*	400	For wider tables

environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n \to \infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath**

environment. Now, we'll enter two unnumbered equations

$$\sum_{i=0}^{\infty} x + 1$$

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_{0}^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

8 CITATIONS AND BIBLIOGRAPHIES

The use of BibTeX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names ("Donald E. Knuth") not initials ("D. E. Knuth") — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the `\end{document}` command:

```
\bibliographystyle{ACM-Reference-Format}
\bibliography{bibfile}
```

where "bibfile" is the name, without the ".bib" suffix, of the BibTeX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the "author year" style; for these exceptions, please include this command in the **preamble** (before "`\begin{document}`") of your LaTeX source:

```
\citestyle{acmauthoryear}
```

Some examples. A paginated journal article [2], an enumerated journal article [7], a reference to an entire issue [6], a monograph (whole book) [19], a monograph/whole book in a series (see 2a in spec. document) [13], a divisible-book such as an anthology or compilation [9] followed by the same example, however we only output the series if the volume number is given [10] (so Editor00a's series should NOT be present since it has no vol. no.), a chapter in a divisible book [30], a chapter in a divisible book in a series [8], a multi-volume work as book [18], an article in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [3], a proceedings article with all possible elements [29], an example of an enumerated proceedings article [11], an informally published work [12], a doctoral dissertation [5], a master's thesis: [4], an online document / world wide web resource [1, 24, 31], a video game (Case 1) [23] and (Case 2) [22] and [21] and (Case 3) a patent [28], work accepted for publication [25], 'YYYYb'-test for prolific author [26] and [27]. Other cites might contain 'duplicate' DOI and URLs (some SIAM articles) [17]. Boris / Barbara Beeton: multi-volume works as books [15] and [14]. A couple of citations with DOIs: [16, 17]. Online citations: [31–33].

REFERENCES

[1] Rafal Ablamowicz and Bertfried Fauser. 2007. CLIFFORD: a Maple 11 Package for Clifford Algebra Computations, version 11. Retrieved February 28, 2008 from http://math.tntech.edu/rafal/cliff11/index.html

[2] Patricia S. Abril and Robert Plant. 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan. 2007), 36–44. https://doi.org/10.1145/1188913.1188915

[3] Sten Andler. 1979. Predicate Path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226–236. https://doi.org/10.1145/567752.567774

[4] David A. Anisi. 2003. *Optimal Motion Control of a Ground Vehicle*. Master's thesis. Royal Institute of Technology (KTH), Stockholm, Sweden.

[5] Kenneth L. Clarkson. 1985. *Algorithms for Closest-Point Problems (Computational Geometry)*. Ph.D. Dissertation. Stanford University, Palo Alto, CA. UMI Order Number: AAT 8506171.

[6] Jacques Cohen (Ed.). 1996. Special issue: Digital Libraries. *Commun. ACM* 39, 11 (Nov. 1996).

[7] Sarah Cohen, Werner Nutt, and Yehoshua Sagic. 2007. Deciding equivalances among conjunctive aggregate queries. *J. ACM* 54, 2, Article 5 (April 2007), 50 pages. https://doi.org/10.1145/1219092.1219093

[8] Bruce P. Douglass, David Harel, and Mark B. Trakhtenbrot. 1998. Statecarts in use: structured analysis and object-orientation. In *Lectures on Embedded Systems*, Grzegorz Rozenberg and Frits W. Vaandrager (Eds.). Lecture Notes in Computer Science, Vol. 1494. Springer-Verlag, London, 368–394. https://doi.org/10.1007/3-540-65193-4_29

[9] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. https://doi.org/10.1007/3-540-09237-4

[10] Ian Editor (Ed.). 2008. *The title of book two* (2nd. ed.). University of Chicago Press, Chicago, Chapter 100. https://doi.org/10.1007/3-540-09237-4

[11] Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies (WOOT '07)*. USENIX Association, Berkley, CA, Article 7, 9 pages.

[12] David Harel. 1978. *LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER*. MIT Research Lab Technical Report TR-200. Massachusetts Institute of Technology, Cambridge, MA.

[13] David Harel. 1979. *First-Order Dynamic Logic*. Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, New York, NY. https://doi.org/10.1007/3-540-09237-4

[14] Lars Hörmander. 1985. *The analysis of linear partial differential operators. III*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. viii+525 pages. Pseudodifferential operators.

[15] Lars Hörmander. 1985. *The analysis of linear partial differential operators. IV*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of

Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. vii+352 pages. Fourier integral operators.
[16] IEEE 2004. IEEE TCSC Executive Committee. In *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*. IEEE Computer Society, Washington, DC, USA, 21–22. https://doi.org/10.1109/ICWS.2004.64
[17] Markus Kirschmer and John Voight. 2010. Algorithmic Enumeration of Ideal Classes for Quaternion Orders. *SIAM J. Comput.* 39, 5 (Jan. 2010), 1714–1747. https://doi.org/10.1137/080734467
[18] Donald E. Knuth. 1997. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.).* Addison Wesley Longman Publishing Co., Inc.
[19] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.
[20] Leslie Lamport. 1986. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, MA.
[21] Newton Lee. 2005. Interview with Bill Kinder: January 13, 2005. Video. *Comput. Entertain.* 3, 1, Article 4 (Jan.-March 2005). https://doi.org/10.1145/1057270.1057278
[22] Dave Novak. 2003. Solder man. Video. In *ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003).* ACM Press, New York, NY, 4. https://doi.org/99.9999/woot07-S422
[23] Barack Obama. 2008. A more perfect union. Video. Retrieved March 21, 2008 from http://video.google.com/videoplay?docid=6528042696351994555
[24] Poker-Edge.Com. 2006. Stats and Analysis. Retrieved June 7, 2006 from http://www.poker-edge.com/stats.php
[25] Bernard Rous. 2008. The Enabling of Digital Libraries. *Digital Libraries* 12, 3, Article 5 (July 2008). To appear.
[26] Mehdi Saeedi, Morteza Saheb Zamani, and Mehdi Sedighi. 2010. A library-based synthesis methodology for reversible logic. *Microelectron. J.* 41, 4 (April 2010), 185–194.
[27] Mehdi Saeedi, Morteza Saheb Zamani, Mehdi Sedighi, and Zahra Sasanian. 2010. Synthesis of Reversible Circuit Using Cycle-Based Approach. *J. Emerg. Technol. Comput. Syst.* 6, 4 (Dec. 2010).
[28] Joseph Scientist. 2009. The fountain of youth. Patent No. 12345, Filed July 1st., 2008, Issued Aug. 9th., 2009.
[29] Stan W. Smith. 2010. An experiment in bibliographic mark-up: Parsing metadata for XML export. In *Proceedings of the 3rd. annual workshop on Librarians and Computers (LAC '10)*, Reginald N. Smythe and Alexander Noble (Eds.), Vol. 3. Paparazzi Press, Milan Italy, 422–431. https://doi.org/99.9999/woot07-S422
[30] Asad Z. Spector. 1990. Achieving application requirements. In *Distributed Systems* (2nd. ed.), Sape Mullender (Ed.). ACM Press, New York, NY, 19–33. https://doi.org/10.1145/90417.90738
[31] Harry Thornburg. 2001. Introduction to Bayesian Statistics. Retrieved March 2, 2005 from http://ccrma.stanford.edu/~jos/bayes/bayes.html
[32] TUG 2017. Institutional members of the TeX Users Group. Retrieved May 27, 2017 from http://wwtug.org/instmem.html
[33] Boris Veytsman. [n. d.]. acmart—Class for typesetting publications of ACM. Retrieved May 27, 2017 from http://www.ctan.org/pkg/acmart

A ERER