

FabTime Cycle Time Management Newsletter

Volume 18, No. 6

December 2017

Information

Mission: To discuss issues relating to proactive wafer fab cycle time management

Publisher: FabTime Inc. FabTime sells cycle time management software for wafer fab managers. New features in development right now include Operation Cycle Time Detail charts and speed enhancements for home page tab management.

Editor: Jennifer Robinson

Contributors: Eric Kerk
(GLOBALFOUNDRIES)

Keywords: Reporting; Metrics and Goals; Fab Management; Factory Ramp-Up

Table of Contents

- Welcome
- Community News/Announcements
- FabTime User Tip of the Month – Sort Stacked Charts by Totals
- Subscriber Discussion Forum
- **Main Topic – Our Required Reading List for New Employees**
- Current Subscribers

Welcome

Welcome to the final issue of Volume 18 of the FabTime newsletter. We hope that you are all having a happy holiday season. In this issue, we are pleased to announce the launch of a new FabTime.com website. We also share a link to some SEMI webinars that we think subscribers may find useful. Our software tip of the month is about sorting stacked bar charts by height. We have two subscriber responses to the last issue, about core practices in transitioning a factory from R&D to production.

In our main article this month, we share FabTime's required reading list for new employees (mainly programmers). We explain our motivation for starting off with these titles, and include brief reviews of all four. We feel that understanding the concepts in these books can make our employees more effective, and at the start of a new year, we would like to share that possibility with our subscribers.

Thanks for reading! We wish you all a joyful holiday season, followed by a happy and productive 2018 – Jennifer and Frank

FabTime

Tel: (408) 549-9932
Fax: (408) 549-9941
www.FabTime.com
Sales@FabTime.com

Community News/Announcements

FabTime Website Redesign

FabTime is pleased to announce a complete redesign of our website, with thanks to Danna Joselovitch of [Danna J. Designs](#). The new FabTime website is streamlined and mobile-compatible, and we hope that you will take a look: <http://www.FabTime.com>.

SEMI Metrics Webinar Education Series for SEMI E10 and E79

We learned recently that SEMI has a series of complimentary webinars through which people can learn more about the widely used industry standards SEMI E10 and

SEMI E79. FabTime has referred extensively to these metrics in our own treatment of availability-related charts in our software, and wanted to bring these webinars to the attention of our newsletter subscribers. The webinars range in length from 30 to 90 minutes, and are presented by David L. Bouldin. More details are available [at the SEMI website](#).

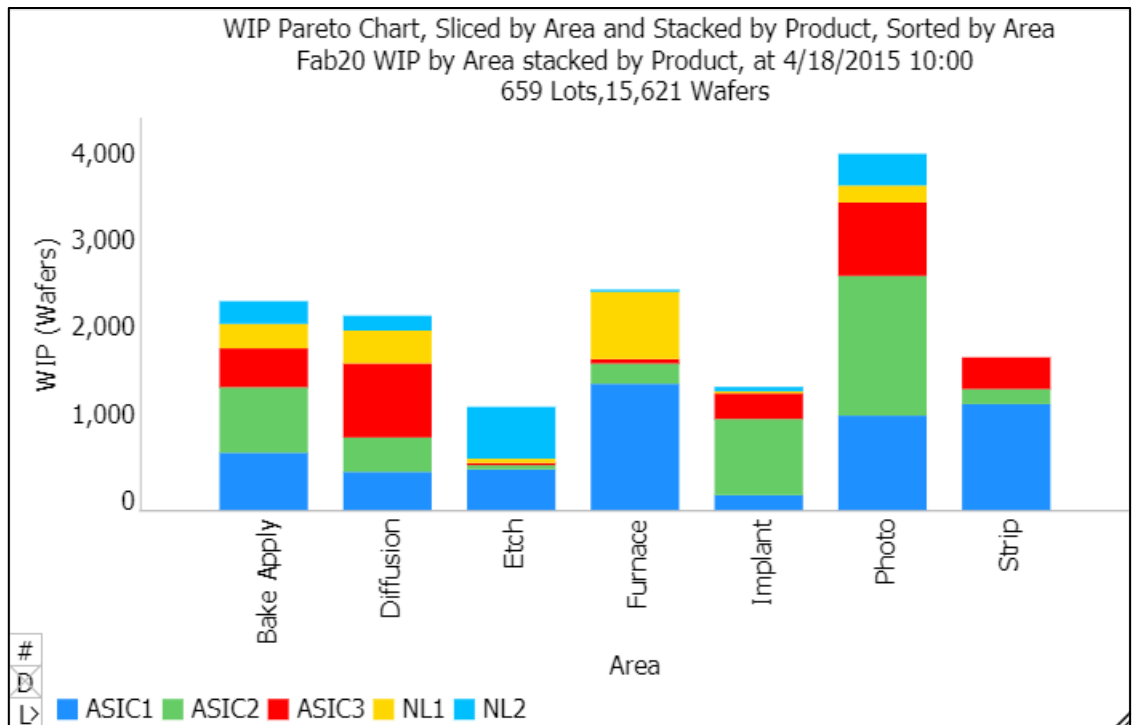
FabTime welcomes the opportunity to publish community announcements, including conference notices and calls for papers. Send them to newsletter@FabTime.com.

FabTime User Tip of the Month

Sort Stacked Charts by Totals

New in Patch108 is the ability to sort stacked charts (trend or pareto) according to the total height of each stacked bar. To do this, first generate the stacked chart of

interest, and set the cross and (if relevant) slice variables. For example, a WIP Pareto Chart, sliced by area and stacked by product, is shown below.



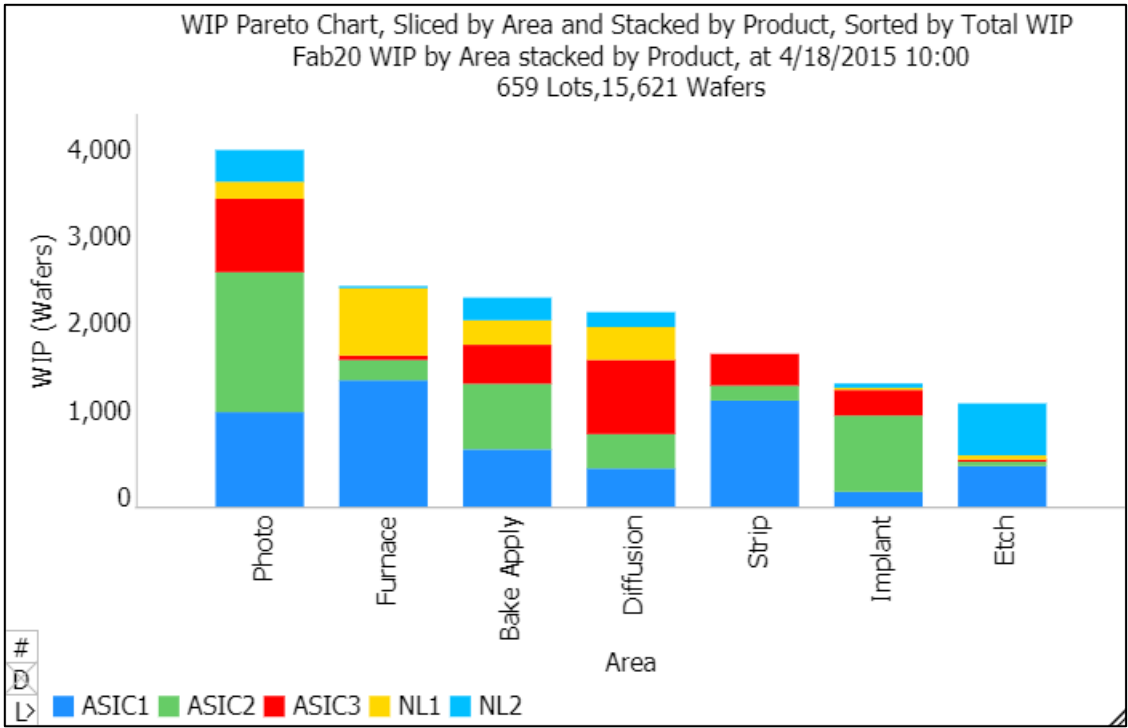
To sort this chart by the total height of each bar, in descending order, use the set of Sort controls to the left of the chart (just below the main set of filters). The first Sort drop-down defaults to “N/A”. Select this drop-down and select “TotalUnits”. Check the box to the right of the field to sort in descending order. Press the “Go” button immediately below the Sort control. FabTime will update the chart, sorting by total stacked bar height, as shown below.

You can of course sort in ascending order by unchecking the box next to that first Sort row. The Sort control does work for Stacked Trend charts, also, though the need for this is likely less common (most

Trend charts are viewed in time order). Note that to sort the rows in the data table in the same manner, you will need to use the separate set of Sort controls located above the data table.

This feature was added at the request of members of the FabTime User Group. As always, we are grateful to the User Group team for helping to make FabTime more useful and user-friendly.

If you have questions about this item, or any other FabTime software questions, just use the Feedback form inside FabTime’s software. Subscribe to the separate [Tip of the Month email list](#) (with additional discussion for customers only). Thanks!



Subscriber Discussion Forum

Transitioning from R&D to Production

In the last newsletter, our main article was about helping factories transition from R&D to production. We received a detailed response from **Eric Kerk (GLOBALFOUNDRIES)** with some suggested additions. Eric wrote:

“Although I am working in a foundry, I can still relate the “Transition from R&D to Production” to the process of qualifying a new product to full production in the foundry.

I would like to add-on to the Core Practice and Key Metrics mentioned in the newsletter.

Core Practice: Mid-Term Planning Meeting

There is one more important meeting held among the Customer Planning, Industrial Engineering, and Production Control/Planning departments. Sometimes, the three roles are merged or overlapped depending on volume or complexity of the production mix; hence I would describe the roles to draw a distinct line between the roles.

1. Customer Planning Dept. – The interface between the customer and Fab. The main interest of this department is to ensure that the Fab fulfills the customer’s demand, in terms of volume or cycle time.

During the transition from R&D to Production, the customer may wish to ramp as fast as possible (steep ramp). However, there are situations that may not be as straightforward as simply ramp or no ramp. Hence, this department needs to ensure that the customer orders are in place at the planned date to start to make sure the ramp materializes.

2. Industrial Engineering Dept. (IE) – The guardian of capacity. The main responsibility of this department is to ensure that each individual Fab module

delivers tool qualifications on time.

Normally, IE will publish a long-term roadmap of the Fab Capacity. However, some organizations may only focus on the Fab Capacity but overlook the progress of the tool qualification. This eventually affects the delivery of the Fab Capacity.

This is absolutely important during the transition period between R&D and Production as R&D may only require only one tool to deliver the customer order. But at production phase, more than one tool is required. Failing to qualify tools in time may result in a delay in product delivery or in yield impact.

3. Production Control/Planning Dept. (PP) – Some people call it the “Brain of the Factory” whereas the counterpart manufacturing team is the “Muscle of the Factory”. PP reads the performance of the Fab and identifies where soft bottlenecks get created in the line. PP makes the adjustment to starts or to balance the WIP. Many points in the newsletter article discussed the roles of PP, hence I would not describe further.

The point that I would like to make here is, apart from the daily reviews and meetings. It is important that the three roles meet and share information on the progress of the ramp from the customer perspective as well as Fab perspective. The shared information will eventually become weekly, monthly and quarterly plans for delivering product throughout the ramp curve (the transition period from R&D to Production). This progress is usually presented bi-weekly or monthly to the management team after the Morning Meeting (Core Practice 1) to align across modules and for the management team to relocate resources if needed.

Key Metrics

Other than those mentioned that are important, Wafer Start and Critical Layer should also be tracked.

Wafer Start

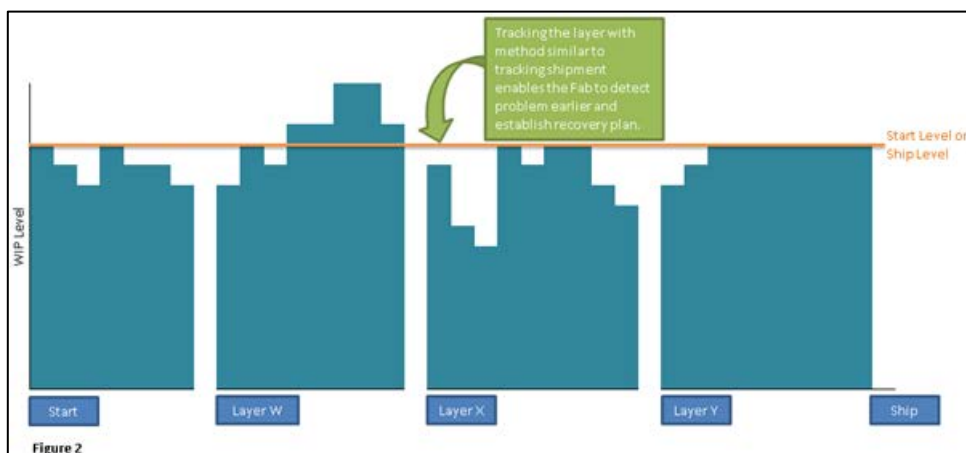
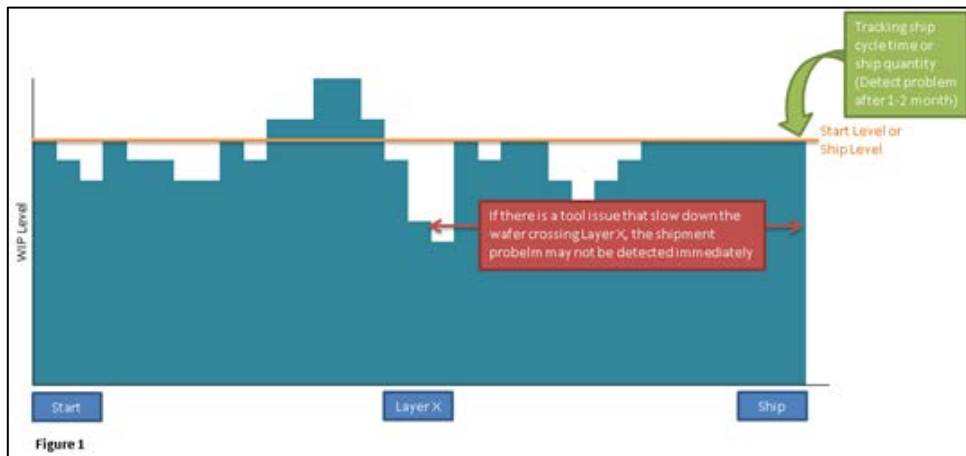
Wafer Start is a number of wafers freshly loaded into the Fab. Other than the total wafers loaded into the Fab, the more important metric is the product mix. During the transition of R&D to Production, if there is only one product, this would not be an issue. However, if the ramp phase involves more than one product, variables are introduced into the Fab. Variables are introduced due to the distinction in Mask Layers or cycle time from Start to Bottleneck among products. Eventually, a soft bottleneck may be created due to the variability and lead to extended cycle time.

Interim Shipments from Critical Layer

The cycle time of a typical Fab is more than 1-2 months. It would be a disaster to discover the Fab is missing the delivery after 1-2 months (see Figure 1, immediately below). In order to detect the problem faster, the wafer process flow is

broken down into zones. The interval between zones is usually 1-2 weeks cycle time. When breaking into zones (Layer W, Layer X and Layer Y), an issue occurring at Layer X would be detected within days instead of week or months (see Figure 2, bottom of page). It is thus valuable to also track “shipments” from each critical interim layer.

FabTime Response: This is an excellent contribution to the discussion, Eric. We especially like your point about the importance of qualifying tools throughout the transition. Our number one recommendation for improving cycle time in fabs is to identify and eliminate single path operations. Unless this is monitored, it would be easy to leave single path operations in place following a ramp to production. The mid-term planning meeting seems like a great addition to our core practices.



As to the addition of product mix, as measured by wafer starts, we think that this is also a good suggestion. Once a fab is running a mature level of production, start changes from week to week tend to not be as critical. But when ramping, one could easily cause serious problems through inattention to the product mix (and impact on things like single path tools).

A number of metrics in FabTime's software allow data to be sliced by segment (what you refer to as Critical Layer). We often recommend looking at WIP by segment, for example, as a measure of line balance. Looking at moves out of each layer does seem like a way to identify throughput problems early, before they flow all the way through the line.

Thank you for these contributions. We look forward to sharing them with other subscribers, and appreciate the chance to learn from your experience.

Another anonymous subscriber also shared some thoughts on hurdles to be overcome in the transition from R&D to production. He noted that retaining an empowered workforce can be challenging because people who had more scope for self-determination in a development factory end up being pressured to do things in a more structured manner, with less freedom to make changes. There's also more pressure in a production environment to meet metrics targets, which can be quite an adjustment. We thought that this was an excellent point.

FabTime welcomes the opportunity to publish subscriber discussion questions and responses. Simply send your contributions to Jennifer.Robinson@FabTime.com.

Our Required Reading List for New Employees

FabTime has hired a number of new programmers over the past few years, many of them starting out with FabTime as summer interns. When we bring these new employees on, we've found it helpful to provide them with a short required reading list to get them up to speed on how we do business. We ask all of our new programmers to read (at least in part) the following four books:

- *The Goal* by Eliyahu Goldratt and Jeff Cox – *The Goal* introduces the terminology our customers use (particularly “bottleneck”), so they (our new employees) will be able to understand more quickly what our customers need.
- *The Effective Executive* by Peter F. Drucker – This book contains the methods we (FabTime) use for running our

business, (we rely on self-directed employees), so that our employees will be able to understand our management practices.

- *The Mythical Man-Month* by Frederick Brooks, Jr. – This book outlines the philosophy we use for software development, so that our new employees will be able to understand FabTime's programming practices.
- *Code Complete* by Steve McConnell - This book is a practical guide to programming that we find helpful in outlining solid general practices for our developers.

FabTime's co-founder and President, Frank Chance, initiated this practice. Frank explains:

“Jennifer and I are both readers. When we started FabTime, we read dozens of books, and shared our notes and recommendations. We focused on areas where we had little experience, e.g. sales, marketing, and management. We were too small to hire an expert in each of these areas, so we had to leverage best practices developed elsewhere. Over the years, we’ve continued reading and sharing notes with each other and with our team. From time to time, we find a new gem, and update our required reading list. These four books form the core of our FabTime technical canon. By “canon,” we mean “The list of works considered to be permanently established as being of the highest quality” (*Oxford English Dictionary*).

I hate to solve problems more than once – there simply isn’t time. At FabTime, we have developed hundreds of checklists for the problems we have solved. But we still find cases where there is no checklist. At these times, I think back to books I have read, to see if there is a similar situation. With these authors, I trust that they have considered the problem carefully and arrived at a well-reasoned solution.

These books provide me with reliable shortcuts to everyday problems. I’d like our employees to use these shortcuts as well. Employees could learn these things piecemeal from me, but they get a complete introduction if they read the original source. And they may discover tips I missed, or have forgotten to mention. On a personal note, I am still working my way through *Code Complete*, and wishing that I had read it years ago. Perhaps that is one mark of a great book – you regret not having read it sooner.

We share brief reviews of each title below.

The Goal

This book, written by Eli Goldratt and Jeff Cox, has been widely read by semiconductor manufacturing personnel since it was first published in 1984. It accurately describes the behavior of

manufacturing facilities, including such fundamental concepts as bottlenecks, constraints, and the impact of variability. One reason why it has been so broadly read is that it frames these concepts in the guise of a novel. This makes the ideas easy to read and digest.

The premise is that Alex, a factory manager, is given an ultimatum – dramatically improve the performance of his factory in three months, or the facility will be shut down. Believing that traditional improvement strategies will never make enough difference in such a short time, Alex must resort to more desperate measures. He tracks down an old professor, now working as a consultant, and begs for advice. The advice of this consultant, Jonah, sets Alex and his team, on a journey. Instead of just giving them the answers, Jonah asks them questions, and refuses to give more help until each question has been answered. As Alex learns through this process, so does the reader.

Some of the lessons of the book include the following.

- When you are productive you are accomplishing something in terms of your goals. Every action that brings a company closer to its goal is productive. The goal of a manufacturing organization is to make money.
- Because of variability, a factory cannot be run at 100% of capacity. Or, as Jonah says, “the closer you come to a balanced plant, the closer you come to bankruptcy.”
- One of the biggest problems in improving your factory is collecting the right data. Alex eventually concludes that “we’re going to have to accept the fact that we’re not going to have perfect data to work with.”
- “An hour lost at the bottleneck is an hour lost for the entire system ... The actual cost of a bottleneck is the total expense of the system, divided by the number of hours the bottleneck

produces.” This suggests managing bottlenecks very closely. This idea has spawned numerous consulting and software firms since the book was published.

- Non-bottlenecks do not need to be regulated so closely, and should not be operated to maximize utilization. Jonah says that “activating a non-bottleneck to its maximum is an act of maximum stupidity.”

FabTime’s co-founders believe that everyone who works in a manufacturing facility should read this book at least once. We re-read it at regular intervals, and always find it insightful.

The Effective Executive

To be effective, Peter Drucker writes, is to do the right thing. This is in contrast to mere busyness—doing many unimportant things right. Effectiveness is a set of habits that can be practiced, and thus learned. Drucker breaks these habits into five categories:

- Know Thy Time
- Focus on Contribution
- Make Strengths Productive
- Do First Things First
- Make Effective Decisions

Drucker begins with time, the fundamental constraint of the executive. He recommends keeping a detailed log of activities, to find out where time is actually being spent. Once this record has been established, it is possible to manage one’s time and to cut back on unproductive usage. The discretionary time that results can then be consolidated into the largest possible blocks. Drucker argues that only with these large blocks of time can the executive tackle the most important tasks and hope to achieve results:

“All one can think and do in a short time is to think what one already knows and to do as one has always done.”

Within each category, Drucker proposes concrete practices that advance executive

effectiveness. He illustrates these practices with examples from his substantial consulting experience. Some of these examples seem dated in this frenzied dot.com era. However, the dot.com executives may be providing us with an updated study on the difference between busyness and effectiveness. As Drucker points out:

“The people who get nothing done often work a great deal harder. In the first place, they underestimate the time for any one task. They always expect that everything will go right. Yet, as every executive knows, nothing ever goes right. The unexpected always happens—the unexpected is indeed the only thing one can confidently expect. And almost never is it a pleasant surprise.”

This book is a concise source of concrete ideas that can be immediately put into practice. For those joining management from the technical world, this book is especially helpful—while you were writing your thesis on superconducting RF cavities, your peers were studying Drucker in business school. If you don’t have a copy, pick one up at the airport before your next flight. Odds are you will dog-ear at least one page from each chapter before you land.

The Mythical Man-Month

Originally published in 1975, by Frederick Brooks, Jr., this book is a classic text on software development. A 1995 edition includes several new chapters of interest, but the original essays remain the heart and soul of the book. In this book, Brooks tackles the question of how to organize and manage large-scale programming projects. These are projects that require hundreds or thousands of programmers, and result in millions of lines of code (think SAP, or Oracle’s database engine, or Microsoft Windows). The book is organized as a series of concise essays.

Brooks’ essays cover a variety of the challenges inherent to large-system

programming, but are useful reading for anyone involved in software development. The name-sake essay (“The Mythical Man-Month”) discusses the indivisibility of many programming tasks, and why this makes the addition of manpower to a software project a futile effort. Frank’s other favorites are “The Tar Pit” (a comparison between the tar pits of prehistory and large-system programming); “Aristocracy, Democracy, and System Design” (a discussion of conceptual integrity); and “Plan to Throw One Away” (on the benefits of explicitly planning for multiple releases before shipment). [Our website review of The Mythical Man-Month](#) includes a detailed analysis of “The Tar Pit” essay as applied to the development of Windows 2000.

Some questions addressed in the book have been made obsolete by technological advances. One example is a discussion of how to distribute typewritten documentation among a large team. You will be surprised, however, at how many of the problems that Brooks faced still frustrate us today. As an added benefit, the brevity and clarity of Brooks’ writing make it a delightful read. If you are a programmer, if you work with programmers, or if you manage programmers, you should read this book.

Code Complete: A Practical Handbook of Software Construction

Frank wrote the first line of FabTime code on November 11, 1999. Not much remains of that first-version code in the product that we ship today. Within our current product, however, there is code written last week, last year, and ten years ago. Small decisions made during coding cast a long shadow, especially in terms of maintenance. Ideally, each line of FabTime code is a) correct, b) efficient, and c) maintainable. As we’ve added programmers, variability in coding style has grown in importance. This variability has a cost when it comes to code maintenance... it’s faster to fix code when

you can read it like a book. It’s easier to read code when it’s written in the style you use every day when writing your own code. The question then becomes – whose style should we use?

Rather than argue, we turn to *Code Complete*. For each of the hundreds of stylistic decisions inherent in any piece of code, *Code Complete* has a reasonable suggestion. Choosing variable names? See Table 11-3, “Sample Naming Conventions” in Chapter 11, “The Power of Variable Names”. Deciding where to place variable definitions, relative to variable use? See Section 10.4, “Scope” in Chapter 10, “General Issues in Using Variables”. It’s like having a friendly chef on call, for every meal-preparation question. *Code Complete* expresses the author’s opinions (“Though sometimes tempting, that’s dumb.”) but in a friendly, I’ve-made-that-mistake-and-you-shouldn’t-too manner.

Code Complete covers an A to Z list of topics for software construction. Due to its daunting length – 900 pages -- we suspect that few programmers read it cover to cover. Unlike a mystery novel, though, it’s possible to read a chapter of *Code Complete*, set it aside for a month, and then pick up with the next chapter. That is the approach we’ve taken at FabTime, and it has paid large dividends. Frank recently revised a significant portion of FabTime’s Javascript chart engine, and used the opportunity to implement recommendations from *Code Complete*. The end result is code that’s easier to read, more likely to be bug-free, and easier to maintain in the coming years.

If you have been programming for many years, you may have little to gain from *Code Complete*. We found quite the opposite – it has spurred us to take a fresh look at our conventions and practices, with an eye toward improving our code quality and readability. We recommend this book for others who believe they could be writing higher-quality code.

Conclusions

When we hire new employees, we often have to spend some time bringing them up to speed on facts about the semiconductor industry. We always, of course, have to introduce them to our software. But we also want to introduce them, as painlessly as possible, to our core beliefs regarding 1) improving manufacturing, 2) running a small business; and 3) writing code.

We could, of course, develop internal training materials to convey this information. But why reinvent the wheel? Eli Goldratt's painless introduction to the basics of factory management stands the test of time, as does Drucker's sage advice about being effective in business. Brooks' philosophical views on programming, combined with McConnell's more pragmatic tactics, combine well to get our programmers into the frame of references that we expect.

FabTime's management team continues to seek out and read other books that we believe will help us to be more effective. For example, our Director of Engineering, Lara Nichols, is currently reading *It's All About Who You Hire, How They Lead...and Other Essential Advice from a Self-Made Leader* (<http://amzn.to/2k96UXo>) by Mort Mandel to gain insights about hiring, and managing teams. Frank, despite having spent more than 30 years programming, added *Code Complete* to his repertoire only recently. Jennifer has been reading books like *A Field Guide to Lies: Critical Thinking in the Information Age* by Daniel Levitin (republished with a new title as <http://amzn.to/2jzKw9B>) and *The End of Average: How We Succeed in a World That Values Sameness* by Todd Rose (<http://amzn.to/2zMDR1P>). We will share these books with our newsletter community as seems useful.

In the meantime, we will continue requiring our new programmers to read the four books discussed above. Even though some are summer interns, not

destined to stay with FabTime long-term, we feel good about making this contribution to their knowledge base. We hope that some of you will find our reading list useful, too.

Further Reading

■ Frederick Brooks, Jr., *The Mythical Man-Month, Essays on Software Engineering, Anniversary Edition (2nd Edition)*, Addison-Wesley Professional, 1995. Purchase from Amazon: <http://amzn.to/2irSqOJ>.

■ Peter F. Drucker, *The Effective Executive (Revised Edition)*, HarperBusiness, 2006. Purchase from Amazon: <http://amzn.to/2hy1a4Z>.

■ Eliyahu Goldratt and Jeff Cox, *The Goal: A Process of Ongoing Improvement (30th Anniversary Edition)*, North River Press, 2014. Purchase from Amazon: <http://amzn.to/2hzgZbF>.

■ Steve McConnell, *Code Complete: A Practical Handbook of Software Construction (2nd Edition)*, Microsoft Press, 2004. Purchase from Amazon: <http://amzn.to/2hHMKmP>.

These reviews, as well as reviews of other books that FabTime recommends, can be found [on our website](#). FabTime is an Amazon affiliate. Purchases made through any Amazon links result in a small commission to FabTime at no additional cost to you. We anticipate using any such commissions to purchase more books.

Closing Questions for Newsletter Subscribers

Does your company have a required reading list for new employees? Or do you perhaps have different lists for different types of employees? Have you read *The Goal*? Have you re-read it in recent years? Are there titles that you think we should add to our canon?

Subscriber List

Total number of subscribers: 2747

Top 20 subscribing companies:

- ON Semiconductor (178)
- Infineon Technologies (146)
- Micron Technology, Inc. (141)
- Intel Corporation (115)
- GLOBALFOUNDRIES (103)
- Maxim Integrated Products, Inc. (99)
- NXP Semiconductors (79)
- Microchip Technology (74)
- Carsem M Sdn Bhd (70)
- STMicroelectronics (64)
- Texas Instruments (61)
- Skyworks Solutions, Inc. (60)
- Western Digital Corporation (57)
- X-FAB Inc. (54)
- Seagate Technology (52)
- TDK (includes RF360) (46)
- Analog Devices (42)
- Honeywell (30)
- ABB (29)
- Silterra Malaysia Sdn. Bhd. (27)

Top 3 subscribing universities:

- Ecole des Mines de Saint-Etienne (EMSE) (15)
- Arizona State University (8)
- Virginia Tech (7)

New companies and universities this month:

- SkyWater Technology (formerly Cypress MN)
- Tech Mahindra Business Services

Sampler Set of Other Subscribing Companies and Universities:

- font
- Acer (1)
- Amkor (2)
- AUO Sunpower Sdn Bhd (1)
- California Polytechnic State Univ. (2)
- Ecole des Mines de Nantes (1)
- eMagin (1)

- Enclos Corporation (1)
- Enovix Corp. (1)
- GKN Driveline (1)
- IQE Semiconductors (1)
- ISS Industrial Services & Solutions (1)
- Kyocera Display Corporation (1)
- LG Display (1)
- Littlefuse (1)
- McKinsey (1)
- Nepes Pte Ltd (1)
- nLight (2)
- Unisem (1)
- The University of Hagen, Germany (1)
- Wichita State University (1)

Note: Inclusion in the subscriber profile for this newsletter indicates an interest, on the part of individual subscribers, in cycle time management. It does not imply any endorsement of FabTime or its products by any individual or his or her company.

There is no charge to subscribe and receive the current issue of the newsletter each month. Past issues of the newsletter are currently only available to customers of FabTime's web-based digital dashboard software or cycle time management course.

To subscribe to the newsletter, send email to newsletter@FabTime.com, or use the form at www.FabTime.com/newsletter-subscribe.php. To unsubscribe, send email to newsletter@FabTime.com with "Unsubscribe" in the subject. FabTime will not, under any circumstances, give your email address or other contact information to anyone outside of FabTime without your permission.

FabTime® Cycle Time Management Software



“Instead of spending time preparing reports, shift facilitators can get the data they need quickly from FabTime, and then spend their time making real improvements.”

Mike Hillis
Cycle Time and Line Yield Improvement Manager
Spansion Fab 25

FabTime Subscription

One low monthly price includes

- Software installation and real-time connect to your MES
- End user and system administrator training
- Unlimited users via your Intranet.
- Software maintenance and regular upgrades (via our no-downtime patch system)
- Add-on dispatching and capacity planning modules for an additional monthly fee

Interested?

Contact FabTime for technical details and/or a web-based demonstration.

FabTime Inc.
Phone: +1 (408) 549-9932
Fax: +1 (408) 549-9941
Email: Sales@FabTime.com
Web: www.FabTime.com

Turn fab MES data into information and save time and money

- Are your supervisors swamped with daily reports, but lacking real-time information?
- Is it difficult to link equipment performance to cycle time?
- Does each new cycle time analysis require IT resources?

FabTime can help. FabTime saves your management team time daily by turning fab MES data into information, via a real-time web-based dashboard that includes lot dispatching. FabTime saves your IT staff time by breaking the cycle of custom-developed reports. With FabTime, the end user can filter for exactly what he or she needs, while staying in a comprehensive framework of pre-defined charts. Most importantly, FabTime can help your company to increase revenue by reducing cycle times up to 20%.

“I use FabTime every day, and so do the supervisors who report to me. The data that I need is right on my home page where I need it when I come in every morning.”

Jim Wright
Production Manager
Headway Technologies



FabTime Benefits

- Cut cycle times by up to 20%.
- Focus improvement efforts on the tools that inflate cycle time.
- Improve supervisor productivity – cut reporting time by 50%.
- Improve IT productivity – eliminate need for custom reports.