IT REVOLUTION IMMERSION

Making Work Visible

Core Concepts Workbook

A COMPANION WORKBOOK TO THE IMMERSION COURSE

DOMINICA DEGRANDIS

25 NW 23rd Pl, Suite 6314
Portland, OR 97210

Cover and book design by D. Smith Creative, LLC

For information about special discounts for bulk purchases or for information on booking authors for an event, please visit our website at www.ITRevolution.com.

MAKING WORK VISIBLE: CORE CONCEPTS WORKBOOK

TABLE OF CONTENTS

About this Workbook

EXERCISE 1 **Value Stream Canvas**

EXERCISE 2 **Setting WIP Limits**

EXERCISE 3 **Dependency Matrix**

EXERCISE 4 **Unplanned vs. Planned Work**

EXERCISE 5 **Conflicting Priorities**

EXERCISE 6 **Create an Aging Report**

ABOUT THIS WORKBOOK

This workbook is designed to help you practice the exercises from the Immersion course Making Work Visible: Core Concepts. I encourage you to use this content in combination with what you have learned from the course videos as well as the primary book for the course, *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*. In addition to instructions to the exercises found in the Immersion course videos, this workbook also includes selected terms and concepts to know as you delve into the exercises. As you conduct each exercise, don't hesitate to go back and rewatch the associated videos and to consult the book as needed.

The Immersion course exercises and the exercises from the *Making Work Visible* book are intended to help you ground the essential, fundamental skills needed to master making your workflow visible. Once you have these skills onboard, you can perform experiments to address pain points, elevate work, and improve your team and your company's desired outcomes.

Let's begin!

EXERCISE 1

Value Stream Canvas

TIME: 60–90 minutes

PURPOSE

To make all the work and handoffs between teams visible as work items flow through your value stream. To gain a shared understanding of the current reality. And to explore areas for improvement.

> Before starting this exercise please be sure to watch Lesson 1 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading the Introduction through Chapter 2.1 from *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*.

DEFINITIONS

- **Value Stream:** The activities done from beginning to end for a specific product or service in order to provide business value.
- **Work Items:** physical or electronic items that track work; encompasses effort both large and small.

MATERIALS

- online tool such as Mural
- or a physical whiteboard w/markers and sticky notes

INSTRUCTIONS

Draw a representation of the activities that occur in your value stream. Include revenue protection activities in addition to revenue generation activities. Show how work arrives and how it is communicated to those impacted. Invite people to talk about handoffs in their workflow—people who participate in the activities necessary to approve, prioritize, design, build, test, deliver, and sustain work.

Identify where work originates from to help everyone see the big picture. Whatever the work item type (incident, problem, defect, etc.), draw a box on your whiteboard or mural canvas to identify the work item and how its arrival is communicated.

Discuss how a typical production issue arrives. Try starting with a production incident because when the production environment is down or unstable, it creates the biggest risk to customer satisfaction and business well-being.

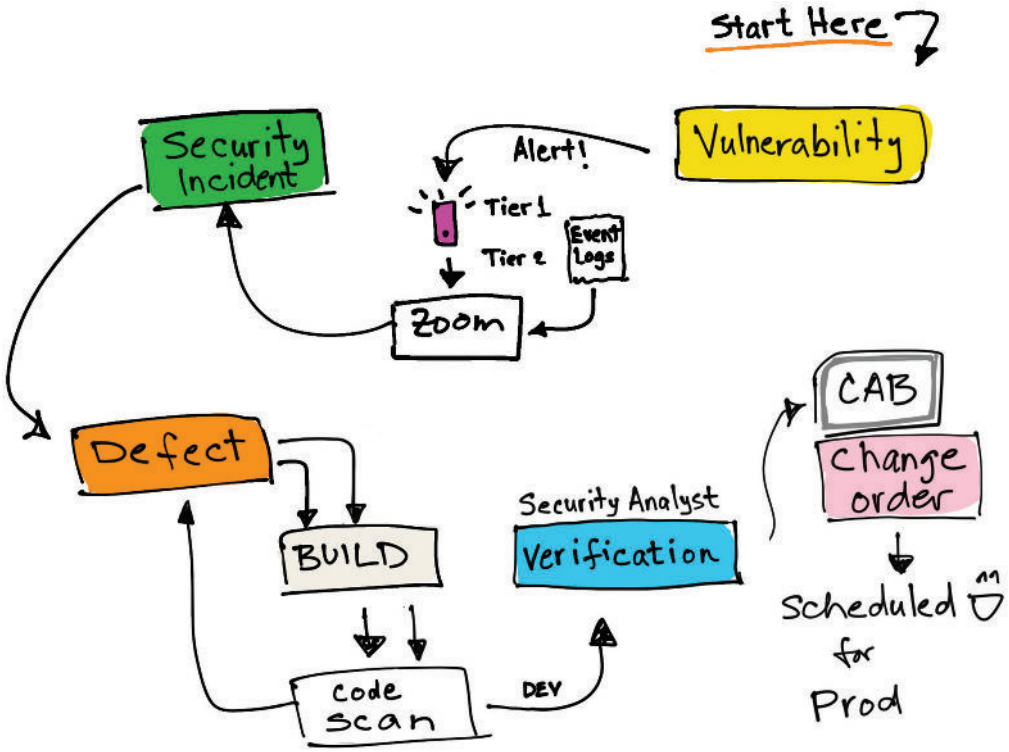


Figure 1: Production Incident Workflow Example

Questions to ask for production workflow:

1. How does a production issue get detected?
2. How does it arrive?
3. Does an alert arrive via a monitoring tool? Does someone email someone?
4. Does someone create an incident or a problem work item in an ITSM tool?
5. What happens next?
6. Who does what and in what order?

Questions to ask for feature request workflow:

1. How does work arrive? Is there a product management org that plans work for a portfolio of products that flows to product development teams?

2. How does your business plan and manage the work? (There's someone out there with a list of things to do—portfolio. Maybe it's in a spreadsheet somewhere.)
3. Does the account management/sales team communicate a new feature request on behalf of the customer? Can customers request changes themselves?

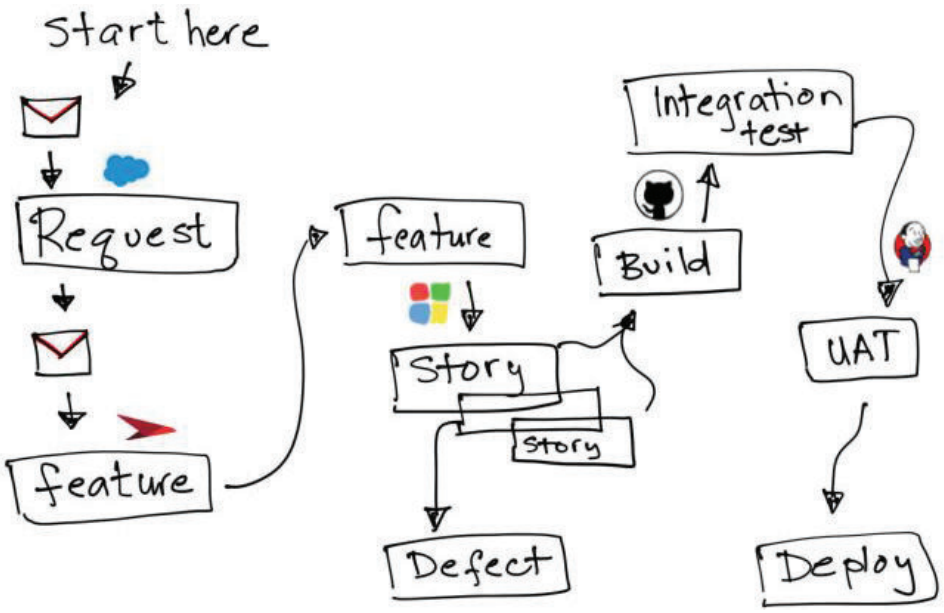


Figure 2: Feature Request Workflow Example

Last, discuss invisible work/wait states in your toolset and what changes you'd advocate for to improve workflow visibility and enable smooth handoffs.

EXERCISE 2

Setting WIP Limits

TIME: 30–60 minutes

PURPOSE

Discuss realistic WIP limits by considering throughput. There is a relationship between the amount of WIP and throughput, and this is what the primary exercise below focuses on. The optional part 2 exercise is for those who want to explore Little's Law.

> Before starting this exercise please be sure to watch Lesson 2 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading about work-in-progress (WIP) from Chapters 1.1 and 2.2 of *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*. Also consider the exercise at the end of Chapter 2.2: Why Take on More WIP.

DEFINITIONS

- **Cognitive load:** The amount of working memory being used.
- **Flow time:** The elapsed time to complete a request from the time work begins to the time the request is available for the customer.
- **Work in progress (WIP):** All the work started but not yet finished.

MATERIALS

- notepad
- a calculator
- input data of 30 days of throughput from your value stream

INSTRUCTIONS

Compare your WIP with your throughput, ideally at the value stream level.

Throughput = Average Number of Items Completed over a Period of Time

Work in progress (WIP) = Average Amount of Work Started but not Finished

Primary Exercise

Identify your average weekly throughput and the current amount of WIP. Divide the WIP by the throughput to get an idea of how many weeks/months it might take to work down your current WIP.

Secondary Exercise: Explore Little's Law Assumptions

There is a relationship between the amount of WIP and flow time—it's called Little's Law, where the average cycle time for finishing tasks is calculated as the ratio between WIP and throughput. WIP is a primary factor in the equation. It's obvious when you think about it—as soon as you get on a clogged freeway, you know that your commute is going to take longer.

Expressed algebraically, the law is:

$$L = \lambda W$$

Where,

W is the Average Mean Time Spent By a Unit in the System

$1/\lambda$ is the Average Number of Items Arriving per Unit of Time

This original formula is written in terms of the arrival rate of work. In knowledge work, however, we usually see Little's Law formula written in terms of the *departure (throughput) rate*:

Flow Time = WIP/Throughput

Where,

Flow Time = average time an item takes to flow through the system

Work in Progress (WIP) = Average Amount of Work Started But Not Finished

Throughput = Average Number of Items Completed over a Period of Time

Little's Law is a relationship of averages. The gist of Little's Law is that—*on average*—the more items that are worked on during the same time interval, the longer it will take to finish those items *on average*. Dr. Little demonstrated that the three key flow metrics—flow time, WIP, and throughput—have a relationship, and why changing any one of these flow metrics can impact the others.

Looking at Little's Law from a throughput perspective brings along with it some assumptions. All metrics are based on assumptions. Little's Law is no different. All you have to do to discredit a metric is to question the assumptions. In order for your metrics to be taken seriously, carefully consider and identify the assumptions in place.

Assumptions about the software delivery workflow process necessary for validity of Little's Law (based on Dan Vacanti's excellent work writing and speaking about Little's Law):

- The average input or arrival rate should equal the average output or departure rate.
- All work that is started will eventually be completed.
- The amount of WIP should be roughly the same at the beginning and at the end of the time interval chosen for the calculation.
- The average age of the WIP is neither increasing nor decreasing.
- Flow time, WIP, and flow velocity must all be measured using consistent units.

The true power of Little's Law is not in calculating the math, but in understanding the assumptions necessary for the law to work in the first place. Using Little's Law to calculate a quantitative forecast is an incorrect application of the law. Little's Law is about examining what happened in the past—not about making deterministic predictions about the future. One cannot predict which of the law's assumptions will be violated in the future and how many times. Each violation of an assumption invalidates the correctness of the law.

Assumption #1: The average arrival rate should equal the average departure rate.

When the arrival rate of work is equal to the departure rate of work, it means that teams can finish work before starting new work, enabling a smooth flow of work through the value stream process. Smooth workflow means less stagnant work and fewer bottlenecks. This is the goal of pull systems and where saying "no" comes into play. Allowing teams to pull work into their work queue based on their available capacity increases the likelihood that the work will have the team's full attention and be processed faster.

Assumption #2: All work that is started will eventually be completed.

This assumption is all about commitment. Committed work means that you have everything you need to do the work. That you have the capability, the approval, the skills, and the money to do the work. Once work passes the line of commitment, the understanding is that it will be worked

on and delivered. Ideally, at this point you could let your customer know the probable timeframe for when the committed item will be delivered.

When teams have high WIP compared to capacity, some work will be neglected and delayed (because teams are busy working on other things), and the likelihood increases that some WIP may be canceled. Teams should be relentless in finishing work before starting new work.

Assumption #3: The amount of WIP is roughly the same at the beginning and the end of the time interval.

Teams that allow WIP to increase over time may notice that Little's Law doesn't work well for them because they're breaking this assumption. Similarly, if a lot of WIP is suddenly removed, the math will break.

Assumption #4: The average age of the WIP is neither increasing nor decreasing.

When average flow time is growing or declining, it means that some items complete faster than other items. This jeopardizes your predictability. It's like robbing Peter to pay Paul, in that you are stealing flow time from other work items that were already in progress.

When work items are "born-in-doing," they skip over older WIP. When that older WIP moves to done, the flow time will be longer than normal because it sat idle and aged. The result is less confidence in the average flow time the team thought they were capable of achieving because past metrics did not include aging WIP.

Also, if the work items that were robbed by Paul see continued neglect, they may end up canceled if they are no longer considered valuable, perhaps because the window of time to realize value has passed. Time matters. When work items fail to complete the process, any effort spent on processing them through the value stream equates to waste.

Assumption #5: Flow time, WIP, and flow velocity must all be measured using consistent units.

This assumption is necessary for the math to be correct. Consistent units of measure are required. This is why story points don't compute when using flow metrics. Because the units of measure for flow time is actual time (days/weeks/months), the throughput measure must also be the same (days/weeks/months).

EXERCISE 3

Dependency Matrix

TIME: 1.5–2 hours

PURPOSE

To surface dependencies in your value stream causing expensive delays because experts aren't available when they are needed.

> Before starting this exercise, please be sure to watch Lesson 3 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading about unknown dependencies from Chapters 1.2 and 2.3 of *Making Work Visible: Exposing Time Theft to Optimize for Work & Flow*.

DEFINITIONS

- **Dependency:** There are three types of dependencies—software/architecture dependencies; people with specialized skill sets needed to do something; and/or an event that needs to occur before something else can be achieved.
- **Theory of Constraints (TOC):** A way to identify the most important limiting factor (the constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.

MATERIALS

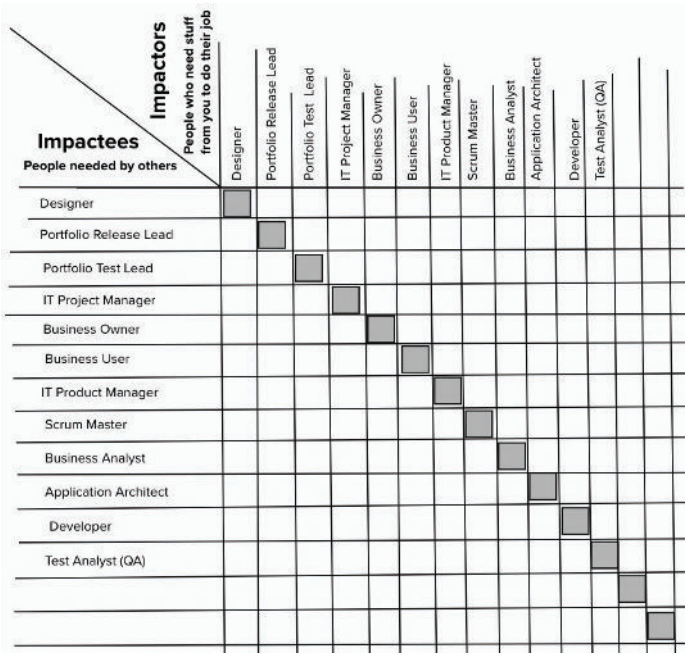
- interactive tool such as Mural (or a whiteboard)
- Dependency Matrix with roles listed (see example below)

INSTRUCTIONS

Step 1 Gather the names of the roles involved in your value stream. Make a list of the roles (similar to this one):

- Systems Analyst
- Desktop Support
- Developer
- Business Owner
- Systems Admin
- Application Architect
- Designer
- IT Project Manager
- Network Engineer
- Scrum Master
- Test Analyst (QA)
- Portfolio Release Lead

Step 2 Add the roles to the vertical column on the matrix. These are the impactees, the people who get interrupted by others who need them to do something. Then copy those same roles to the horizontal lanes, which represent (the impactors), people who need someone else to do something so they can finish their job.

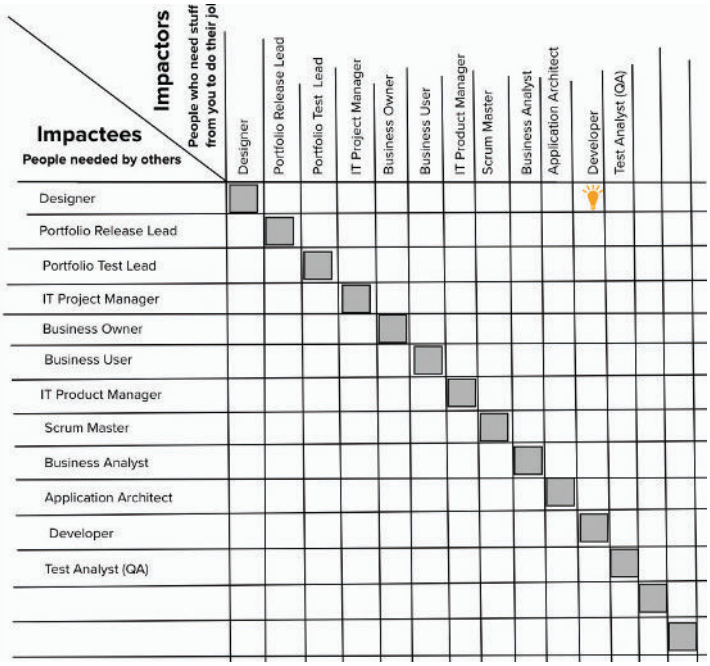


For each role listed in the column to the left of the matrix (the impactees), discuss who they get interrupted by. Keep in mind that we're not trying to identify every dependency here, just those with the highest delays

Visualize the most critical delays by adding an icon (orange light bulb in the next example, but you could add a clock icon) to the intersecting squares of the Dependency Matrix to represent delays.

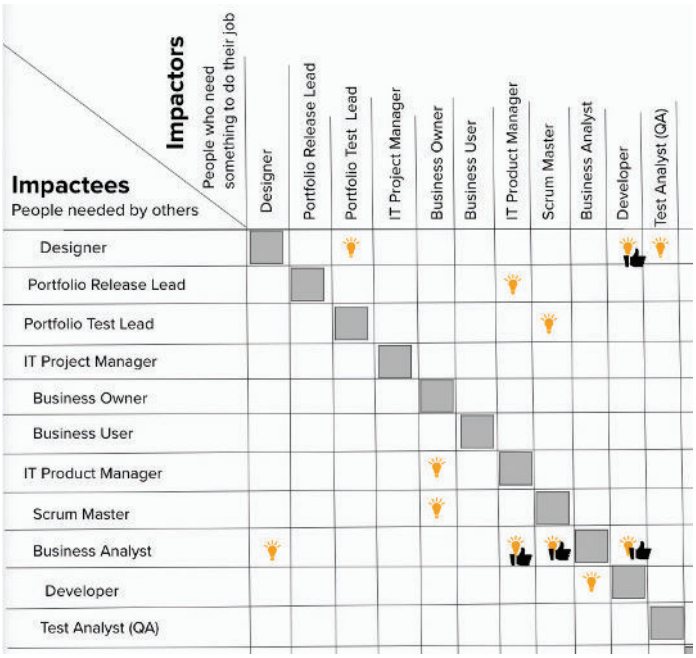
This can be confusing. Just remember that when you turn your head sideways to read the names listed horizontally across the top of the matrix, those are the people who need others to do something so they can finish their job.

If a developer needs wireframes to do their job, then they have a dependency on a designer, and you would place an icon here:



Step 3 Continue to go down the list of roles and surface expensive dependencies (e.g., surface orange) due to expertise/specialization delays.

Step 4 Ask people to add an icon (thumbs up for example) to those dependencies that they believe are the biggest bottlenecks in their end-to-end workflow



Step 5 Discussion topics:

1. What actions can be taken to reduce risk of negatively impacting another team's work?
2. Do your current workflow states provide sufficient visibility on wait states due to dependency on experts?

EXERCISE 4

Unplanned vs. Planned Work

TIME: 60 min

PURPOSE

To make unplanned work visible so that it can be measured. Measuring the ratio of planned work vs. unplanned work prompts conversations for change and helps teams take actionable steps to improve problems related to unplanned work.

This exercise is good to do with a small team. Invite those who are involved and impacted by unplanned work and who have access to tools where unplanned work data resides.

> Before starting this exercise please be sure to watch Lesson 4 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading about work in progress (WIP) from Chapter 1.3 and Chapter 2.4 from *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*. Consider also the exercise at the end of Chapter 2.4, The Interruption Reduction Experiment.

DEFINITIONS

- **Unplanned work:** Work that catches you off guard; work you can't anticipate.

MATERIALS

- The amount of planned and unplanned work over the last 60–90 days.

INSTRUCTIONS

Step 1 Clarify how your team defines "unplanned work." What does unplanned work mean in your organization? What are the criteria? Responsible communication requires us to clearly define what we're talking about. Does your unplanned work come in the form of rework, expedited work requests, or special VP needs? Is it unplanned because production is down? Or because another team thinks their request is of higher priority than your current work?

Here are some examples of unplanned work. Your context may be different:

- break-fix work
- rework from poor process
- special VP requests
- "quick question" emails
- urgent audit documentation
- organizational changes
- searching for information

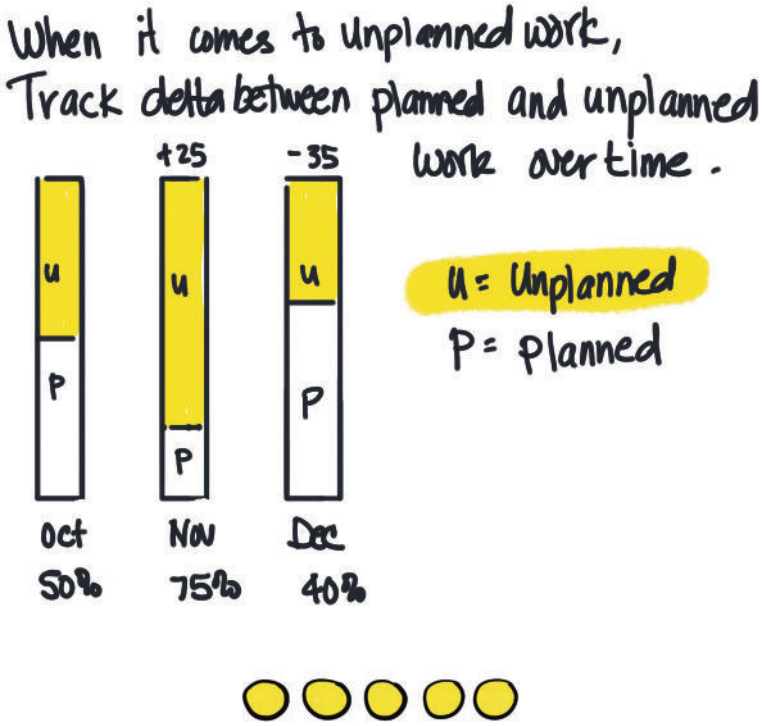
Step 2 Discuss ways to make unplanned work visible in your toolset. Unplanned work may show up in different work item types—from features to defects to technical debt. What options do you have to flag or tag work items that are unplanned?

Step 3 Now that you've made unplanned work visible, measure the trend of unplanned work vs. planned work over time. Calculate your ratio of planned work vs. unplanned work over the last 2–3 months. Do this by dividing the number of unplanned work items completed in a month by the total number of planned and unplanned items completed in that month. Perhaps a query on flagged fields in your workflow tool can aid you in this effort.

A = # of Unplanned Work Items Completed

B = # of Planned and Unplanned Work Items Completed

A/B = Ratio



EXERCISE 5

Conflicting Prioritization Exercise

TIME: 60 minutes

PURPOSE

Make conflicting priorities visible to expose causes of friction. Show work delayed from conflicting priorities across competing projects.

> Before starting this exercise please be sure to watch Lesson 5 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading about conflicting priorities from Chapters 1.4 and Chapter 2.5 of *Making Work Visible: Exposing Time Theft to Optimize for Work & Flow*. Consider also the exercise at the end of Chapter 2.5, Visualize Priorities.

DEFINITIONS

- **Cycle time:** A measure of speed or the elapsed time it takes to complete a request from the time the work began to the time the work was delivered.
- **Flow distribution:** The ratio of different types of work completed.

MATERIALS:

- whiteboard or online tool such as Miro or Mural to aid further discussion through visual means.

INSTRUCTIONS:

Review the outcomes of the Dependency Matrix exercise from lesson three of this workbook. Invite impactees (those needed by others) to identify common conflicting priorities. Create a table to visualize concurrent projects and requests resulting in conflicting priorities.

List causes of conflicting priorities. For example, a designer may receive wireframe or illustration creation requests from four different requesters or projects at the same time. Or triage and/or planning efforts that compete with each other during project ramp up and ramp down cycles.

These can be listed out in a simple table, like the one provided below:

	Triage	Plan	Design	Validate
Project X	✓	✓	✓	
Project Y	✓		✓	
Project Z		✓	✓	✓
Project Q	✓		✓	
Project T	✓	✓		✓

Discuss the Following

1. How do people working on multiple requests decide what work to do next?
2. What types of requests tend to get delayed (or dropped)?
3. Measure the cycle time of requests resulting in conflicting priorities. Review which states in your value stream are active vs. in a waiting state? If that capability doesn't exist yet, research and discuss how to enable cycle time metrics so you can compare them with your end-to-end flow time of work across your value stream.

If managing work by product, review the flow distribution of work, where

Features = Revenue Generating Biz Requests (Epics/Features/Stories)

Defects = Defects, Bugs

Risks = Security Compliance, Audits, Gov't Mandates, Taxes, GDPR

Debt = Investments in Future Delivery Capability (Architecture, Tools, People, or Process)

Discuss the Following

1. What was the intended distribution?
2. What were the tradeoffs with this distribution?
3. Retro your prioritization decisions.

EXERCISE 6

Create an Aging Report

TIME: 60 minutes

PURPOSE

To improve the flow of value by making neglected work visible. An aging report shows WIP that is stale.

> Before starting this exercise, please be sure to watch Lesson 6 of the Immersion course Making Work Visible: Core Concepts. We also recommend reading about neglected work from Chapters 1.5 and Chapter 2.6 from *Making Work Visible: Exposing Time Theft to Optimize for Work & Flow*. Consider also the exercise at the end of Chapter 2.3, The "Oh, By the Way" Dependency Matrix.

DEFINITIONS

- **Technical Debt**: Extra effort required to fix software bugs and develop new features because of previous quick and dirty design choices.

MATERIALS

- Interactive tool such as Mural (or whiteboard)
- Your current workflow management toolset.

INSTRUCTIONS

Query your work-tracking tool to find high priority, yet partially completed work that has not moved or been updated in 30 days. If 30 days results in a huge list, then increase to 60 or 90 days. Randomly select 7 to 11 of those high priority items. Statistically, 7 to 11 is sufficient as long as it's truly a random sample set.

For each of the seven to eleven items, note the following:

- The number of days the item has been stale (not updated or moved or made in progress).

- The average cycle time for items of a similar work item type.
- How many days the item has been in progress in comparison to other similar work item types.

Now, write down what happens if this item continues to be delayed for another week. Consider lowering your WIP limit and reprioritizing your WIP based on what might happen if the work is delayed. Identify the most valuable work currently in progress and separate out the lower value items. If possible, allocate additional capacity for an improvement effort to push through the one highest priority item to get it delivered. Improve flow by making important but stale neglected work visible.

ABOUT THE AUTHOR

A huge fan of using visual cues to inspire change, **Dominica DeGrandis** is one of the IT industry's top workflow experts. She is currently Principal Flow Advisor at Tasktop and is the author of *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*. Obsessed with elevating the state of work, Dominica advises customers on value stream management, flow metrics, and how to affect change in their organizations by making their work visible across value streams. You can find her on LinkedIn at www.linkedin.com/in/dominicadeg and Twitter at @dominicad.