|  |
| --- |
|  |
| **Team Foundation Server** |
| **Practical Kanban Guidance** |
|  |
| Wednesday, July 18, 2012 |
| **Visual Studio ALM Rangers** |
| **Microsoft Corporation** |

|  |
| --- |
| **Visual Studio ALM Rangers**  This content was created by the Visual Studio ALM Rangers, a special group with members from the Visual Studio Product Team, Microsoft Services, Microsoft Most Valued Professionals (MVPs) and Visual Studio Community Leads. |

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft grants you a license to this document under the terms of the Creative Commons

Attribution 3.0 License**.** All other rights are reserved.

2012 Microsoft Corporation.

Microsoft, Active Directory, Excel, Internet Explorer, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

[Foreword by Gregg Boer 6](#_Toc330394517)

[Introduction 7](#_Toc330394518)

[Overview 7](#_Toc330394519)

[Visual Studio ALM Rangers 7](#_Toc330394520)

[Understanding the Personas and Scenarios 8](#_Toc330394521)

[Overview 8](#_Toc330394522)

[Personas 8](#_Toc330394523)

[Bob 8](#_Toc330394524)

[Jane 9](#_Toc330394525)

[Graham 9](#_Toc330394526)

[How to use this Guidance (Scenario Mapping) 10](#_Toc330394527)

[Scenarios 10](#_Toc330394528)

[Scenarios and Guidance Cross Reference 10](#_Toc330394529)

[Summary 10](#_Toc330394530)

[Introduction to Kanban 11](#_Toc330394531)

[What is Kanban? 11](#_Toc330394532)

[Principles of Kanban 11](#_Toc330394533)

[Visualize the Workflow 11](#_Toc330394534)

[Limit Work In Progress 12](#_Toc330394535)

[Manage the Flow of Work 12](#_Toc330394536)

[Benefits of Kanban 13](#_Toc330394537)

[When to use Kanban 13](#_Toc330394538)

[How to get started 13](#_Toc330394539)

[Implementing Kanban with Team Foundation Server 14](#_Toc330394540)

[The Microsoft Kanban 1.0 Process Template 18](#_Toc330394541)

[Deploying the Microsoft Kanban 1.0 Process Template 18](#_Toc330394542)

[Deploying the Global List Synchronizer 19](#_Toc330394543)

[Creating a New Team Project with the Microsoft Kanban 1.0 Process Template 20](#_Toc330394544)

[Available Work Item Types 24](#_Toc330394545)

[Setting up the Process Definition 25](#_Toc330394546)

[Tracking a Card through Its Life Cycle 28](#_Toc330394547)

[Queries 33](#_Toc330394548)

[Reports 34](#_Toc330394549)

[GlobalListUpdaterPlugin detail 34](#_Toc330394550)

[Overview 34](#_Toc330394551)

[Setup 36](#_Toc330394552)

[Administration 38](#_Toc330394553)

[Migrating Existing Projects to the Microsoft Kanban 1.0 Process Template 40](#_Toc330394554)

[Adding the New Work Item Types to an Existing Team Project 40](#_Toc330394555)

[Adding the New Queries to an Existing Team Project 40](#_Toc330394556)

[Install the Global List Updater 41](#_Toc330394557)

[Adding the New Reports to an Existing Team Project 42](#_Toc330394558)

[Next Steps 44](#_Toc330394559)

[Managing and Tracking Progress with Kanban 45](#_Toc330394560)

[Measurements in Kanban 45](#_Toc330394561)

[Interpreting the Kanban Board report 45](#_Toc330394562)

[Interpreting the Cumulative Flow Diagram report 47](#_Toc330394563)

[Further Reading 50](#_Toc330394564)

[Books 50](#_Toc330394565)

[Blogs 51](#_Toc330394566)

[Web Sites 52](#_Toc330394567)

[Other Resources 52](#_Toc330394568)

**Table of Figures**

[Figure 1 Card State Model 26](#_Toc316213875)

[Figure 2 Kanban State Global Lists 27](#_Toc316213876)

[Figure 3 Process Steps Query 27](#_Toc316213877)

[Figure 4 Editing a Process Step 28](#_Toc316213878)

[Figure 5 Global List Synchronization 29](#_Toc316213879)

[Figure 6 Creating a New Card 30](#_Toc316213880)

[Figure 7Taking a Card From The Backlog 31](#_Toc316213881)

[Figure 8 Completing a Stage 32](#_Toc316213882)

[Figure 9 Queued Cards Query 33](#_Toc316213883)

[Figure 10 Completing a Card 34](#_Toc316213884)

**Table of Tables**

[Table 1 - Scenarios to guidance mapping 10](#_Toc311646252)

# Foreword by Gregg Boer

My name is Gregg Boer and I work in the product group of the Team Foundation Server team. Specifically, I’m the Program Manager for the Agile management tools we are releasing in this upcoming release. I am also planning how we will support Lean and kanban going forward.

I worked with the team that produced this guidance and the Microsoft kanban 1.0 Process Template. A lot of thought went into deciding what to do. In fact, the first three sections detail out the problem space and the analytical process that was followed.

In short, the team followed a few key principles:

* It must be easy for a customer to try out
* It must be easy for a customer to customize their kanban board.

If you want to jump right into what was produced, skip to the “The Microsoft Kanban 1.0 Process Template” section.

As a member of the product group, one key thing I am looking for is feedback from actual customers using kanban tools to manage their project. This feedback is invaluable to us; because it helps the product group develop a solution that meets real customer’s needs.

No one on the team believes that this solution is the full solution to all-things-kanban, but we do firmly believe that getting a solution out there for people to use, and then listen to their real-world feedback, is more valuable than all the focus groups in the world.

After you’ve had a chance to try it out, and have feedback on what would make a great Kanban tool, please feel free to reach out to me on twitter (@greggboer), or my email [gregg.boer@microsoft.com](mailto:gregg.boer@microsoft.com).

Gregg Boer

Principal Program Manager

Team Foundation Server

# Introduction

## Overview

This guidance is not an introduction to Kanban, nor does it offer guidance on how to use Kanban, we offer pointers in the Further Reading section if you need this sort of guidance. This guidance is only intended to help you use Team Foundation Server as your tooling to support your Kanban project.

This guidance delivers practical and scenario based guidance for the implementation of Kanban using Team Foundation Server. We start with a brief description of Kanban, and then move on to provide you with guidance on how to structure a Team Project for Kanban and what reports you can get from Team Foundation Server to support your use of Kanban.

## Visual Studio ALM Rangers

Visual Studio ALM Rangers is a special group with members from the Visual Studio Product group, Microsoft Services, Microsoft Most Valued Professionals (MVP) and Visual Studio Community Leads. Their mission is to provide out of band solutions to missing features and guidance.

This guide is intended for Microsoft “200-300 level” users of Team Foundation Server. They are intermediate to advanced users of Team Foundation Server and have in-depth understanding of the product features in a real-world environment. Parts of this guide might be useful to novices and experts, but they are not the intended audience for this guide.

# Understanding the Personas and Scenarios

WHAT’S IN THIS CHAPTER?

* Definition of **personas** that are targeted by this guidance.
* Mapping of personas and **scenarios**.

## Overview

This guidance is based on hypothetical personas, and scenarios. The intention is to demonstrate, in a realistic and convincing way, how the different personas can use Kanban in their teams, making use of their investment in Team Foundation Server to help them manage their processes.

## Personas

The three personas interested in this guidance are focused on planning, setup and day-to-day management of the Team Foundation Server environments.

### Bob

** Bob** is the senior developer on a team that currently uses a waterfall process to deliver enterprise financial software within a bank. The team is made up of developers, an architect, Claire, and a project manager, Dave.

The team releases software every quarter and the content of the release is agreed and planned in advance between the Dave, Claire, and the business representative. They typically have a successful release each quarter but it usually requires long hours for the last few weeks. The business would like changes more often, but understand the amount of effort required in getting a release out.

Analysts create requirements that are passed on to the team but these often have too little details or too much detail. Often, Bob has to spend a lot of time clarifying the requirements and usually, by the time Bob gets round to writing the code, the real requirement has changed. This frequently results in a change request from the business that often delays the requirement until a subsequent release. This frustrates the business.

When a release is code complete, it is deployed into a test environment where the test analysts test the new version before it is packaged and deployed. There are usually a lot of bugs reported by the test analysts and this often delays a release.

Bob has read a lot about agile and has discussed it many times with Claire, Dave and the other developers. They understand that agile methods would help them produce their software quicker and without so much rework, but when they have tried adopting it in the past, the other teams of analysts and testers haven't wanted to get involved and make changes to their processes.

Recently, Bob attended a presentation on Kanban and heard how it allows teams to gain a lot of the advantages of agile by starting with what they do now and steadily and continually improving.

Bob wants to know how they can get started with Kanban using their Team Foundation Server 2010 system.

### Jane

** Jane** is a Scrum Master on a team that currently uses a Scrum process to deliver a social networking application in "the cloud". The team is made up of 3 developers, 2 testers and 2 designers.

They've been using Scrum for 2 years, and release updates to the application usually every month. As well as working on new features they also have to maintain the existing application and develop and deploy bug fixes. Jane finds that the random nature of the bugs distracts the team from their current Sprint commitments. She's found it difficult to explain to the owner of the business, Fiona, that the bugs should just be added to the next sprints backlog as she wants them fixed as soon as possible.

At the end of each sprint, the team always identifies some great improvements to their process but they typically find that they never have time to actually implement them, features or bug fixes always take higher priority. This is frustrating for the team.

Jane knows that Scrum works, they've been using it to deliver successfully for 2 years, but she is struggling to figure out how they can ship features, fix production bugs, implement all the great improvement ideas AND keep Fiona happy.

Jane has read that Kanban can help with this and wants to understand how they can use Kanban with their Team Foundation Server.

### Graham

** Graham** is the project manager of a large team (40+) at an insurance software ISV. They started using Kanban 6 months ago as a way to reduce the lead time for new features on their flagship product, InsureULike.

The development team is divided into feature teams that look after different parts of the product. They have separate User Experience and Database teams, that are shared across the feature teams.

Graham tracks all work using a Kanban board on the office wall but has recently been having problems with this. They first started using sticky notes but found that they would fall off the board overnight and they didn't know where they were in the process flow. They switched to paper cards and pins which solved this problem. Graham updates a CFD and SPC chart manually in Excel, then prints them out each day and sticks them to the board.

About a month ago, they expanded the team by adding a new feature team in Bengaluru, India. Unfortunately, they can't see the board or update their work on the board.

Furthermore, as they are a financial software company they need to keep an audit of their work.

The team already uses Team Foundation Server 2010 for source control.

Graham is interested in how Team Foundation Server could be used to solve his Kanban problems.

## How to use this Guidance (Scenario Mapping)

### Scenarios

This guidance addresses the following scenarios:

* Understand what Kanban is about and why we should consider it.
* How to structure team projects and Work Item Types to support Kanban
* How to visually track my team progress with Team Foundation Server

### Scenarios and Guidance Cross Reference

| Scenario | Refer to | Bob | Jane | Graham |
| --- | --- | --- | --- | --- |
| I would like to understand what Kanban is about and why we should consider it. | Introduction to Kanban  page 11 |  |  |  |
| I would like to know how to structure team projects and Work Item Types to support Kanban | Implementing Kanban with Team Foundation Server  page 14 |  |  |  |
| I would like to know how to visually track my team progress with Team Foundation Server | Managing and Tracking Progress with Kanban  page 36 |  |  |  |

Table - Scenarios to guidance mapping

## Summary

This guidance is based on the following personas:

**Personas**

* Bob, the development lead who is currently still using waterfall.
* Jane, the Scrum Master currently using Scrum.
* Graham, the project manager who is already using Kanban manually.

# Introduction to Kanban

WHAT’S IN THIS CHAPTER?

* What is Kanban?
* Principles of Kanban
* Benefits of Kanban.
* When to use Kanban.
* How to get started.

WARNING

This guidance is on how to use Team Foundation Server as your tooling to support your Kanban project. It is not intended as a complete introduction to Kanban and how to use it, for that you are referred to the Further Reading section on page 37.

## What is Kanban?

Kanban is a process improvement method. In software development, Kanban can be applied to your existing software development process to help you to improve it. Kanban draws on the principles of Lean Manufacturing, which itself draws on the discipline of Systems Thinking, to take an overall view of how your current software development process is working end-to-end, and to identity where wasteful activity is occurring.

Wasteful activity is often quite difficult to see, because the people involved in it are usually very busy, making them appear to be productive in the eyes of the management; but often the wrong thing is being measured, so the people *are* busy, but they are not always being productive in terms of the real end product.

Sometimes, the people involved in wasteful work are just too busy to have the time to reflect on the real value of the work they do; or they may have only a localized view of their part of the process, and feel that, because they are busy, they are adding value. More often, people will have an intuition that something is not quite right in the way they are having to work, but lack the authority or influence to do anything about it.

Kanban provides a way for these inefficiencies to become apparent, so that you can then act to improve the process, and at the same time remove the frustrations felt by many of the people involved in the process.

## Principles of Kanban

In the software development world, Kanban focuses primarily on the time it takes for work to go through your software development process, from conception to final delivery to the customer, known as the Lead Time. The delivery to the customer represents realized value and so by reducing the lead time the customer is receiving value more quickly. The principles that help you to achieve this are described in the following sub-sections.

### Visualize the Workflow

To benefit from Kanban you must first understand the flow of work through your software development process, and identify the process steps in your process (e.g. Analyze, Design, Develop and Test). You then need to find a way to visualize this flow and, in particular, where all your units of work are in the flow.

This is usually done with a Kanban Board. There are many variations, but most are in essence a set of columns across the board that represent the process steps of your workflow, with cards on the board representing each unit of work you are tracking.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backlog | Analyze | Develop | Test | Done |
| [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) | [Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) |

### Limit Work In Progress

Kanban asks you to limit the amount of Work In Progress (WIP) at each stage of the work flow. The intention of this is to emphasize the completion of current work over the acceptance of new work. This has the effect of allowing people doing the work to concentrate on fewer things at the same time. This means they are not constantly switching from one piece of work to the next, which is wasteful. It also allows the work to get through each stage more quickly. Limiting WIP also reduces a buildup of unfinished work, allowing work, once started, to get through the process more quickly.

Implementing WIP limits requires work to be pulled into each process step, rather than pushed. A step of the process is not allowed to start a new piece of work if it is already at its WIP limit. Only when a unit of work leaves a step, so that the step is under its WIP limit, can the next piece of work be pulled in. This prevents the buildup of work in any one process step. This buildup represents inventory which a lean process seeks to minimize. This cascades through the workflow to limit the amount of WIP in the whole process.

Limiting WIP also means that the output of each stage of the process is a queue of work ready to go to the next stage of the process. The crucial element here is that the work in a stage’s output queue counts to that stage’s WIP limit. This is done to prevent the buildup of an inventory of unfinished work, which leads to various inefficiencies. It is quite possible for a stage to have completed all its work and to be idle, even if the previous stage has work ready for it. When this happens it is telling you that you have a bottleneck in your process, and a process improvement opportunity.

WIP is represented visually on the typical Kanban Board, by splitting the columns for each stage in the flow of work into two sub-columns, one for work actually in progress, and one for work that has been completed, but not yet taken into the next stage. The WIP Limit is usually written above the two sub-columns.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Backlog | Analyze (3) | | Dev (4) | | Test (3) | | Done |
|  | Analyze | Queue | Develop | Queue | Test | Queue |  |
| [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) |  | [Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) |  | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) |  | [Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010Thumbnail for version as of 13:59, 16 August 2010](http://upload.wikimedia.org/wikipedia/commons/e/e5/Post-it-note-transparent.png) |

### Manage the Flow of Work

Visualizing the flow of work through the process and preventing the buildup of unfinished work are both important. However, the ultimate aim is to improve the process. The status of the workflow must be monitored, measured and reported, and as problems become apparent they can be addressed so that the software development process you are using is continuously improved.

Kanban teams typically measure the Lead Time for each unit of work that flows through the process.

## Benefits of Kanban

Being a method for improving a process, Kanban does not require you immediately to change your processes and whole way of working. You can take an evolutionary path rather than a revolutionary one, which decreases resistance to change and increases the likelihood of a positive outcome for you. Also, by reducing waste, people doing the work will be more productive and feel much more valuable, valued and involved, which also serves to improve morale and wellbeing.

Kanban gives your team a much deeper understanding of your current software development process and the information you need to improve that process. Your team will process each unit of work more quickly and efficiently. You will be able work with your customer on a more frequent basis to prioritize which unit of work to start on, the next time you have the capacity. Your process will become much more predictable, giving your customer confidence in your prediction of when the unit of work will be delivered to them. Being in a position to allow your customer frequently to choose what work gets done next, with a predictable lead time, is very valuable to most customers.

## When to use Kanban

Kanban is an effective choice for improving a teams’ software development process. Being able to implement Kanban without having to immediately your processes provides a low barrier to entry. By mapping out the value stream of our existing software development process and understanding and improving bottlenecks, incremental improvement can be realized without the impact of wholesale change.

Kanban is simply a process improvement technique or method. If your team is looking to make continuous improvements to their process then Kanban can help enable that improvement.

## How to get started

In order to implement Kanban there are some simple steps to follow:-

Don’t change anything – The start point for Kanban is your existing process. Improvements will emerge over time. It is important to begin using the current process. By supporting the existing process no initial change is required to adopt Kanban. This is why Kanban is an incremental improvement framework.

Map the Value Stream / Visualize the workflow – The existing process should be displayed visually. Kanban builds on Visual Management techniques. Constraints are often easy to identify when the entire process is visible. A Kanban board is often used for this with a number of columns representing the process steps within the overall process.

Limit the Work in Progress – Each process step is assigned a WIP limit. This ensures that the process does not become blocked with too many tasks. Limiting the amount of work in progress helps to identify constraints and bottlenecks within the process.

Measure – Lead time is measured to understand the elapsed time between the customer requesting change and that change being deployed. Kanban is focused on minimizing the lead time.

Change – Based on evidence from measurements and the impact of the current WIP limits the process can be changed. This can include changing the WIP limits at the process steps. The impact of these changes is measured to ensure that the results are positive.

Loop back to Measure ad inifinitum – Kanban is an evolutionary process. The quest for continuous does not end.

# Implementing Kanban with Team Foundation Server

WHAT’S IN THIS CHAPTER?

* What is needed from a process template to model Kanban in Team Foundation Server.
* How Kanban can be modeled in a process template.
* Description of the process template accelerator.

Requirements

There are a number of requirements that need to be satisfied to allow teams to get the most out of a Kanban-enabled process template for Team Foundation Server. These are described below.

The process template should make it easy to modify the states in the process flow.

Teams need to be able to decide how they wish to work. One team may want a process flow that starts with an analysis phase, followed by implementation and testing phases before the card is complete Another team may wish to add a user experience design phase to the lifecycle of a card.

Furthermore, Kanban is about continuous improvement. During the life of a project a team may wish to modify their flow, adding or removing states.

Therefore the process template should make it easy to add and remove states in the process flow.

The process template should allow Work In Progress limits to be represented and reported on.

One of the few fundamental concepts in Kanban is to limit the amount of work in progress (WIP). This means that the process template should allow us to specify a WIP limit for each state in the lifecycle of a card.

We also need the team to be able to easily see how many cards are in each state and to see if any WIP limits are being exceeded. However, it is not necessary to implement a system that prevents WIP limits being exceeded.

The process template should allow each state in the process flow to have an In Progress and Done sub-state.

When running a process using Kanban each state in the process flow has an In Progress and a Done sub-state. The Done sub-state is actually the queue into the next state of the process flow. WIP limits apply to the whole state. For example, if the Implementing state has a WIP limit of 5, this means that if there are 3 cards that have been done and have not yet been taken into the Test stage of the flow, then only 2 cards are allowed to be In Progress in the Implementing state.

The process template should allow for In Progress and Done sub-states and report on WIP limits across the two sub-states. The process template should not require the author of the process flow to have to define the sub-states, these should be automatically provided.

The process template should support arbitrary state transitions.

Kanban is not a software development method and does not dictate anything about how a team should implement its process. Therefore a card should be allowed to be moved into any Kanban state from any other Kanban state. This means that the team is in control of the process rather than the process template, and it allows the team to work in the way that best helps to get the work done.

The process template should support hosted Team Foundation Server instances.

Increasingly, software development organizations are providing centrally managed Team Foundation Server services to their software development teams. This means that teams are not fully in control of their Team Foundation Server.

To give teams maximum flexibility, a Kanban-enabled process template should not require project collection administration rights to make changes such as the states in the process flow and WIP limits.

Inevitably though, Kanban-enabling a process template does require the template to be changed, but this should be a one-time only step. Projects can then control the process flow and WIP limits with no more than project administration rights.

The process template should not break existing reports.

One of the strengths of Kanban is that it does not require existing processes to be changed. It allows process improvement to be introduced in an evolutionary manner, rather than creating a revolution, which inevitably increases resistance and the likelihood of failure.

Modifying an existing process template to enable Kanban should not therefore affect the existing reports. This helps to minimize any possible resistance to change coming from outside the team which wishes to introduce Kanban.

Of course, over time the existing reports may lose their significance, as process improvements are introduced, but it means that resistance to the introduction of Kanban is minimized because the existing reports continue to work.

The process template should include a Cumulative Flow Diagram Report

The process template should include a Cumulative Flow Diagram report. This makes it easy to see when bottlenecks are occurring, and measures lead time. This allows the team to monitor how it is doing and if it is succeeding in improving its processes.

The process template should include a Kanban Board Report

The process template should include a report which shows a Kanban Board so that the team can see at a glance what the state of the work is, where there is queuing and how it is doing with respect to the established WIP limits.

The process template ought to allow different process flows in the same Team Project for different teams.

Different teams in a project may have a requirement to have a different process flow for their cards. This conflicts somewhat with the concept of a Team Project implementing a single process, but in a large project it is conceivable that different Areas may have slightly different processes. For example there might be a team working on the batch system and another team working on the web site, their Kanban process flows could well differ.

Ideally, therefore, a Kanban-enabled process template should allow different process flows in the same Team Project, allowing each Area to have its own. This is considered a less important requirement.

Representing Kanban State in a Process Template

There are a number of ways that the Kanban state of a card could be represented in a Process Template. This section outlines those methods, presents the advantages and disadvantages of each when assessed against the requirements presented above, and finally presents the chosen method, explaining the reasons for the choice.

Method 1. Use Iterations to Represent Kanban State.

This method uses a distinct iteration for each Kanban state. It is a simple method that does not require any modifications to existing process templates. However, iterations have a specific meaning in Team Foundation Server, and in the forthcoming release their meaning will become even more significant as they will have a start and end date. This is not a recommended solution.

Method 2. Use Work Item States.

In many ways this is the most obvious method because work items already have a state. It would be a matter of creating work item types with states that reflect the chosen Kanban states. This has a greater impact on a Process Template because if affects any existing reports and queries.

Adding or removing a state requires the work item state, and allowed transitions to be changed. With arbitrary state transitions required it can be difficult to manually add a new state, so a tool would be needed to generate the relevant definitions.

Method 3. Use Dedicated Fields for Kanban State.

In this method two new fields are added to the work item types that are tracked on a Kanban board. The first is the Kanban state (e.g. Analyze, Design, Implement, or Test). A second field indicates if the state is in progress or done. The list of allowed Kanban states is set in a global list.

A dedicated work item type is used to store metadata about the Kanban states, in particular the current WIP limit for the state. So there is one instance of the item for each Kanban state. The global list of states can be synchronized with the work item instances using a service.

One detail about this solution is that global lists are common across an entire team project collection, to accommodate different lists for different projects, a naming convention for the global lists can be used, so that different lists are used in different team projects.

Comparison of Methods

| Requirement | Method 1 Iterations | Method 2 Work Item State | Method 3 Dedicated Fields |
| --- | --- | --- | --- |
| The process template should make it easy to modify the states in the process flow. | Just add an iteration. Requires administrative permissions. | Change state, but complex state transition maintenance needs tool support. Also needs administrative permissions. | Add or remove instance of special work item type. No special permissions needed. |
| The process template should allow Work In Progress limits to be represented and reported on. | External mechanism required. | External mechanism required. | Represented in the special work item types. |
| The process template should allow each state in the process flow to have an In Progress and Done sub-state. | External mechanism required. | Yes, but proliferates even more states. | Second field used to do this. |
| The process template should support arbitrary state transitions. | Yes | Yes, but complex to manage, needs a tool to generate the work item type definition. | Yes |
| The process template should support hosted Team Foundation Server instances. | Yes | Only if process template customization is allowed. | Only if process template customization is allowed. |
| The process template should not break existing reports. | Breaks existing reports | Breaks existing reports | Existing states unaffected so reports will work, but many will have reduced meaning. |
| The process template should include a Cumulative Flow Diagram Report | Yes | Yes | Yes |
| The process template should include a Kanban Board Report | Yes | Yes | Yes |
| The process template ought to allow different process flows in the same Team Project for different teams. | Yes | No | Possible if the special work item type includes a field for the area it applies to. |

Choice of Kanban State Representation

The method that uses iterations can be discarded immediately because iterations have another meaning entirely in Team Foundation Server, and this method is not recommended.

Using work item state is possible and is a reasonable method, but suffers from some complexity in maintaining the states and impacts existing projects much more.

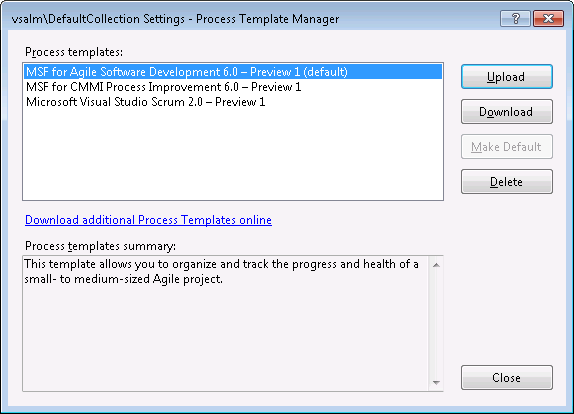
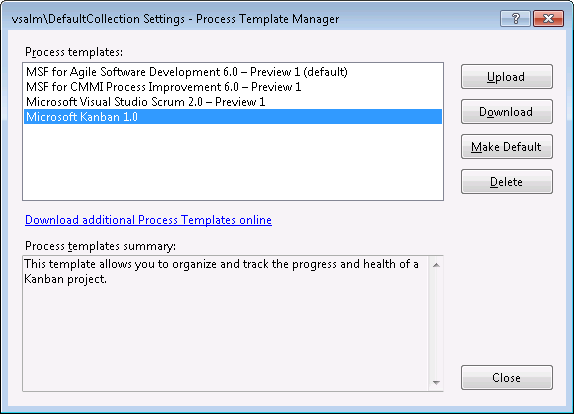
Ultimately, the method using dedicated fields is the most flexible, and once the process template has been modified to support Kanban, the team does not need further administrator intervention to modify the process the team uses. This is the method of choice.

## The Microsoft Kanban 1.0 Process Template

This section describes the Microsoft Kanban 1.0 Process Template that comes with this guidance. The template is provided in two versions, one for Team Foundation Server 2010 and one for Team Foundation Server 2012.

### Deploying the Microsoft Kanban 1.0 Process Template

Follow these steps to install the Microsoft Kanban 1.0 Process Template. The steps are shown using Team Foundation Server 2012, but are almost identical to those required for Team Foundation Server 2010.

1. Download the template from <http://vsarkanbanguide.codeplex.com/> and copy the files to a local directory.
2. Start Visual Studio 2010 or later.
3. From the **Team** menu select **Team Project Collection Settings -> Process Template Manager…**. You will see a dialog like the following.  
     
   
4. Click **Upload** and browse to the folder where you saved the process template files. This is the folder where the file ProcessTemplate.xml is stored. Once the folder has been selected Visual Studio will upload the template and when this has completed the template is available to create new Team Projects. Following a successful upload you should see the following list:  
     
   

### Deploying the Global List Synchronizer

The Global List Synchronizer makes administration on Kanban States easier and is described later. This section describes how to deploy it. Follow these steps:

1. Download and compile the GlobalListUpdater as described in The GlobalListUpdaterPlugin described in detail on page 37.
2. Log in to the Application Tier server.
3. For Team Foundation Sever 2010: Copy the compiled assemblies from step 1 to <System Drive>:\Program Files\Microsoft Team Foundation Server 2010\Application Tier\Web Services\bin\Plugins.
4. For Team Foundation Server 2012: Copy the compiled assemblies from step 1 to <System Drive>:\Program Files\Microsoft Team Foundation Server 11.0\Application Tier\Web Services\bin\Plugins.
5. If you have multiple Application Tiers, repeat step 2-4 for each of them.
6. Make sure that the service account used by Team Foundation Server has permission to update global lists.

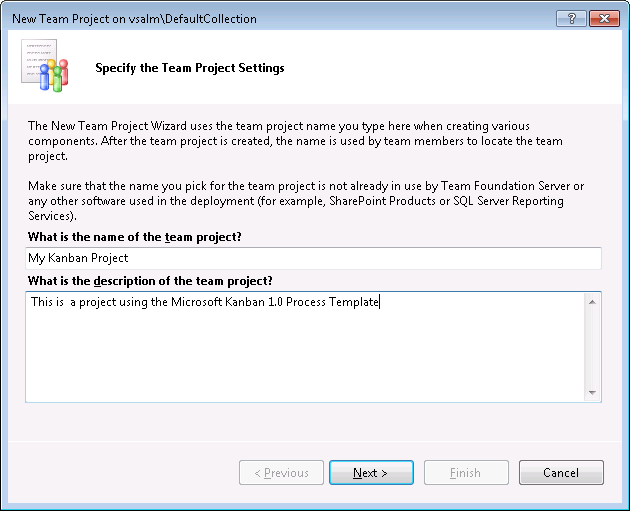
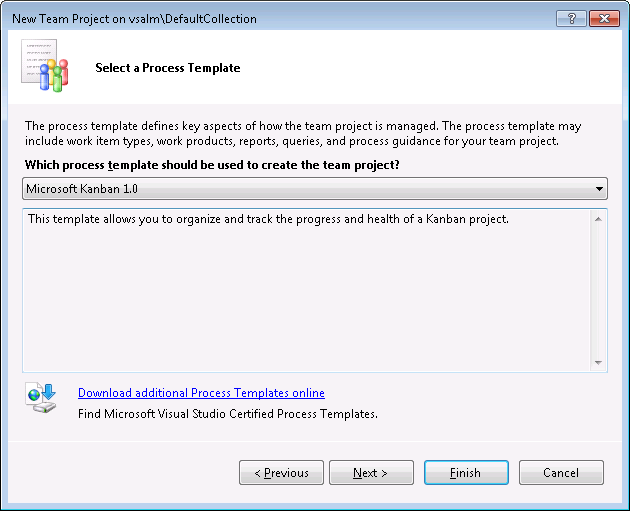
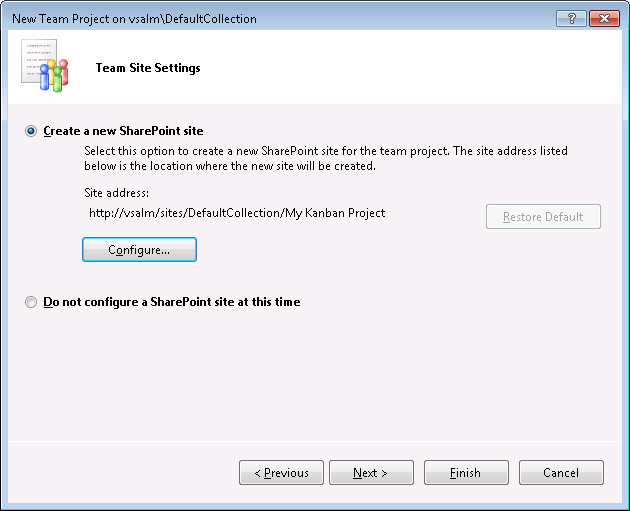
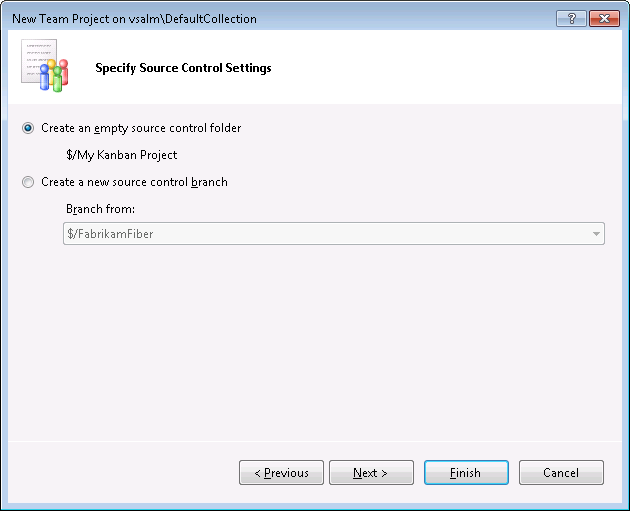
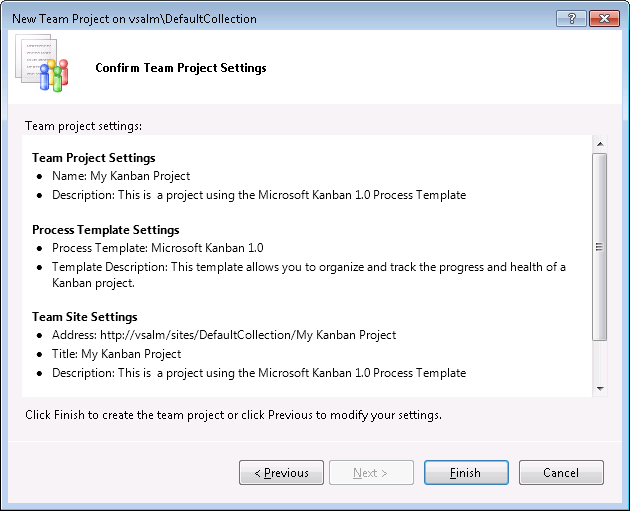
The Plugin has a configuration file which you can use to customize for your specific needs. If you use the Microsoft Kanban 1.0 Process Template, no changes to the configuration are necessary. The comments in the configuration file explain the configuration settings, should you need to change them.

The GlobalListUpdaterPlugin writes messages to the Application Event Log for errors as well as for information. If you update your Process Steps work item and don’t get the expected result, make sure you check the Event Log.

You should install and configure the GlobalListUpdaterPlugin before you create your first Team Project because only then it will create the Global List entries and change the WITD to use the correct Global List immediately.

### Creating a New Team Project with the Microsoft Kanban 1.0 Process Template

Once the process template has been installed you can create a new Team Project based on the template. Create a Team Project in the normal way as follows:

1. Start Visual Studio 2010 or later.
2. From the File menu choose **New -> Team Project**., this will display the following dialog:  
     
   
3. Enter the name and an optional description for the Team Project and click **Next**. This will display the following dialog:  
     
   
4. Select the Microsoft Kanban 1.0 Process Template and click **Next**. This will display the following dialog:  
     
   
5. Select **Create a new SharePoint site** and click **Next**. This will display the following dialog:  
     
   
6. Choose your desired source control settings and click **Next**. This will show the final dialog:  
     
   
7. Review the settings and once you are satisfied click **Finish**. Visual Studio will then create the new Team Project.

Once the Team Project has been created remember to set up the security on your Team Project. This is beyond the scope of this guidance.

Available Work Item Types

The Process Template provides the following Work Item Types in addition to the Work Item Types that come with the MSF for Agile Process Template (except for User Story which is replaced by Card):

1. **Card**. This is the unit of work that is tracked through your software development process. A Card represents value demand or failure demand on your team. You can specify a type of Card. By default the following Card types are provided, although you can customize the card types by changing the Kanban Card Types global list.
   1. **Work Task**. This represents demand on your team for work that has value for your customer (value demand).
   2. **Team Task**. This represents demand on your team for work that has intangible value. Typically these includes work such as setting test environments, refactoring some code etc.
   3. **Problem**. This represents demand on your team to rectify a problem (failure demand).
2. **Process Step**. This Work Item Type is used to define information about the steps in your process. In particular this is where the WIP limits are defined for each step of your process.

Setting up the Process Definition

To use the Microsoft Kanban 1.0 Process Template, it is important to define the software development process as a linear sequence of steps. The card state model is shown below:



Figure Card State Model

The software development process is defined using the Process Step Work Item Type, which is used to define the Configurable Kanban States shown in Figure 1. One work item is defined for each step in your process. Figure 2 below illustrates how the Process Step work items are used to populate a Global List for each Team Project, the Global List is used to define the allowed Kanban States in a Team Project, so that every Team Project can have its own list of Kanban States. The Global List Synchronizer (described later) implements this process:



Figure Kanban State Global Lists

The process template pre-defines some Kanban States for you. Running the Process Steps query (in the Process query folder) will show you the following pre-defined states:

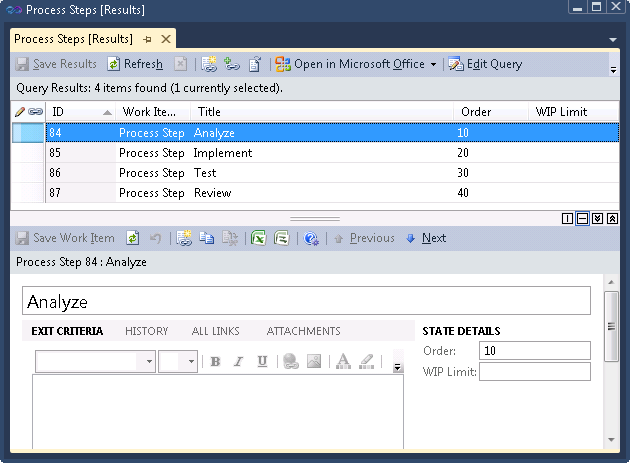


Figure Process Steps Query

If all you need are those steps then just edit the WIP limit and you are ready to go. Adding a new process step or editing an existing one is straightforward, an example is shown below:

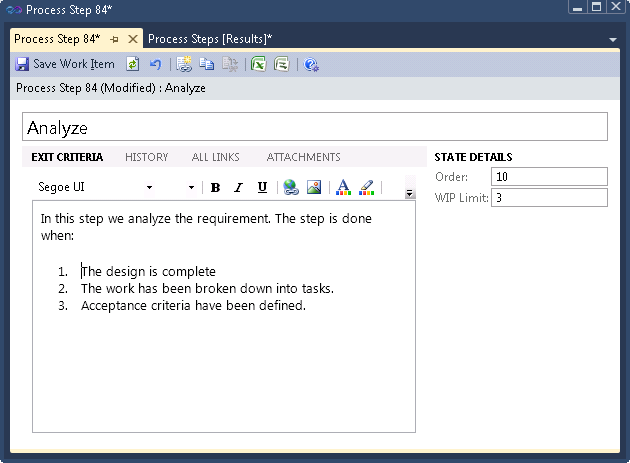


Figure Editing a Process Step

In the above example the Analyze process step is being edited. It has a WIP Limit of 3. The Order field is the position of this process step in the overall software development process.

A background process called the Global List Synchronizer, synchronizes the list of process steps used in the Card Work Item Type with the current list of Process Steps. As you add, change or remove Process Step work items, the list of valid process steps in the Card work items changes accordingly. It modifies the Card work item type definition so that each Team Project uses its own Global List to provide the list of process steps. The following diagram shows how the background process works:



Figure Global List Synchronization

### **Tracking a Card through Its Life Cycle**

#### Creating a Card

The life of a card starts by creating a Card Work Item, which puts it on the project’s backlog. You create the Card in the same way that any Work Item is created. To create a Card enter the following information:

1. The title of the card and a description.
2. Planning information.
3. The Card Type, this indicates the kind of demand it represents on the team.

As it is on the backlog it should not be assigned to anyone. The State is Planned and the Kanban State is Backlog., these together represent the Card on the backlog. An example of a new card is shown below:

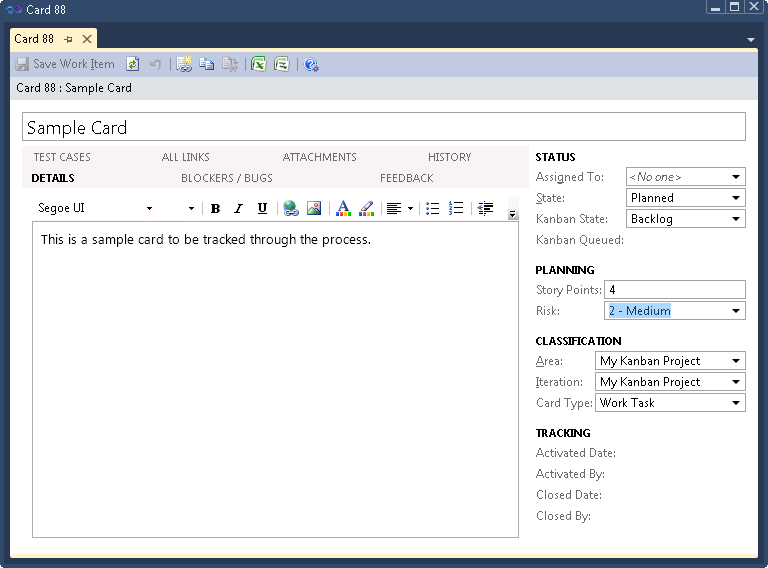


Figure Creating a New Card

#### Taking a Card from the Backlog

When the amount of WIP in the first stage of your process falls below the WIP limit, you can take a Card off the backlog. To take a Card off the backlog, do the following:

1. Set the Card State to In Progress. The card remains in this state throughout the time it is active in the process.
2. Set the Kanban State field to the first stage of your process.
3. Set the Assigned To field to the person who is now dealing with the Card.

**Do not** set the State to Complete when a person has done their part of the work, the Complete State is set only when the Card has completed its passage through the entire process.

Taking a Card off the backlog will change the Kanban Queued field to No, which means that the Card is currently being worked on in the assigned stage. This also establishes the start time for the Card (Activated Date field), which is used to calculate the lead time for the Card when it completes its cycle through the process. An example is shown below:

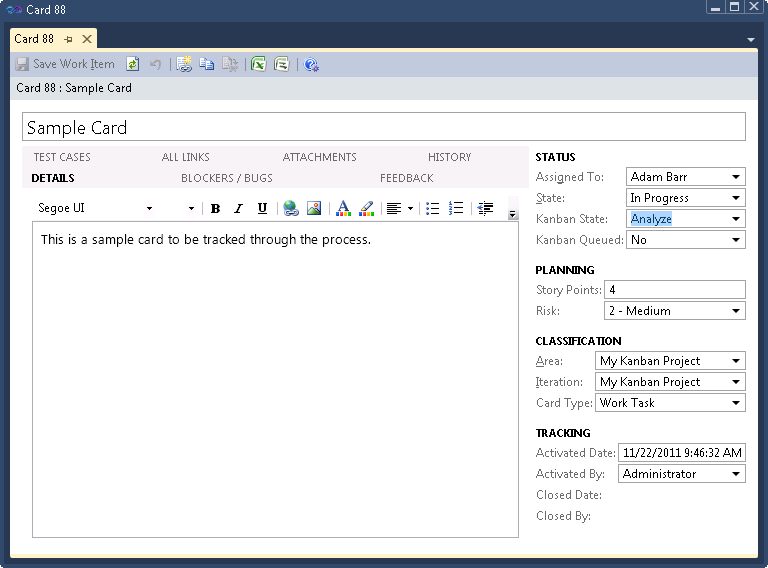


Figure Taking a Card from the Backlog

#### Completing a Stage

When the person responsible for working on a Card has completed their work and has satisfied the project’s definition of done for that state, the following steps must be carried out:

1. Change the Kanban Queued field for the Card to Yes.
2. Set the Assigned To field to be blank.
3. **Do not** change the State field, it must remain In Progress because the Card has other stages of the process still to go through.
4. **Do not** change the Kanban State field, the Card is still considered to be in the current stage for WIP limit purposes.

The Card is now in the queue for the next stage of the process. However, it still counts towards the WIP limit for the stage that it is in. The Card is ready for the next stage to pull it in, when that stage has the capacity available.

An example of a Card that has completed a step in the process is shown below:

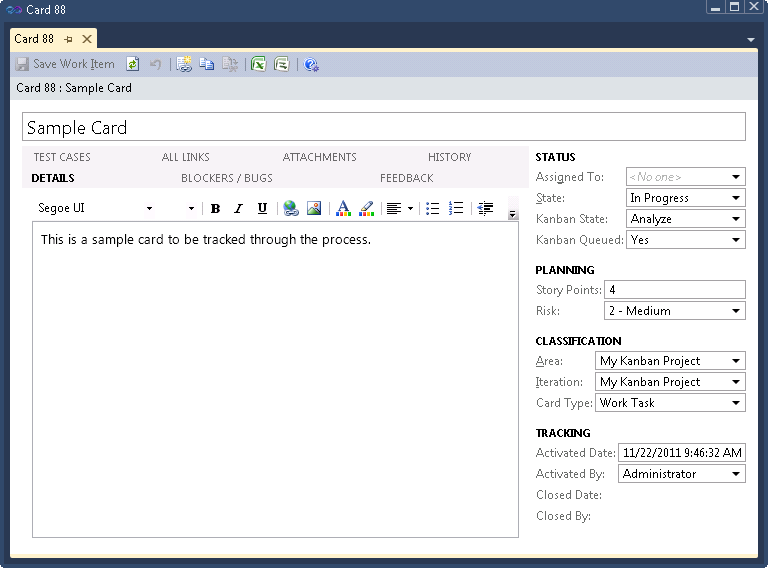


Figure Completing a Stage

#### Moving a Card to the Next Stage

When the amount of WIP in a stage falls below the WIP limit that has been set for the stage, a Card can be taken from the queue of Cards that have completed the previous stage. To view the queue execute the query called Queued Cards, an example is shown below:

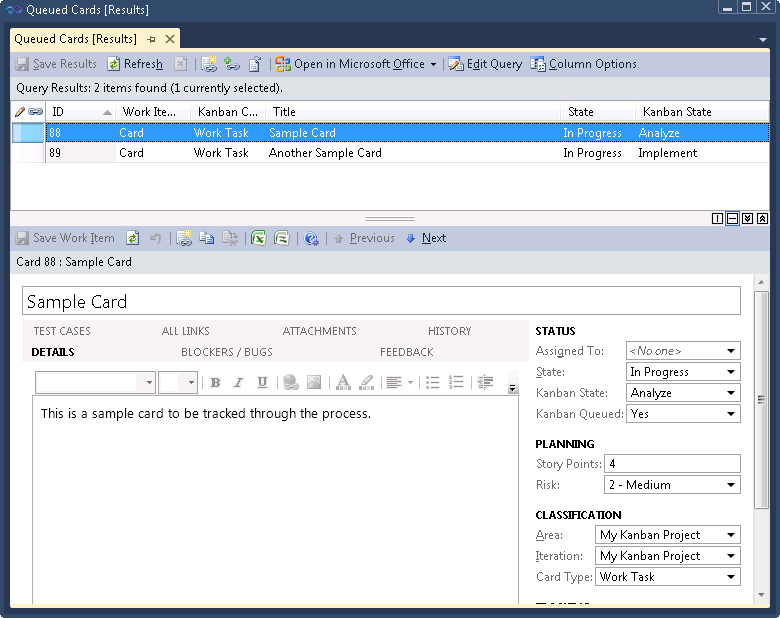


Figure Queued Cards Query

Follow these steps to move a card into the next stage.

1. Set the Kanban State field to the next stage of your process. This will automatically set the Kanban Queued field to No.
2. Set the Assigned To field to the person who is now dealing with the Card.

#### Completing the Process for a Card

When a Card has completed the final stage, having met all the project’s definitions of done, do the following:

1. Set the Card State to Complete.
2. Set the Assigned To field to be blank.

This sets the Kanban State field to Closed. Doing this also records the time the Card was completed in the Closed Date field, and so allows reports to calculate the lead time for the Card. An example of a completed Card is shown below:

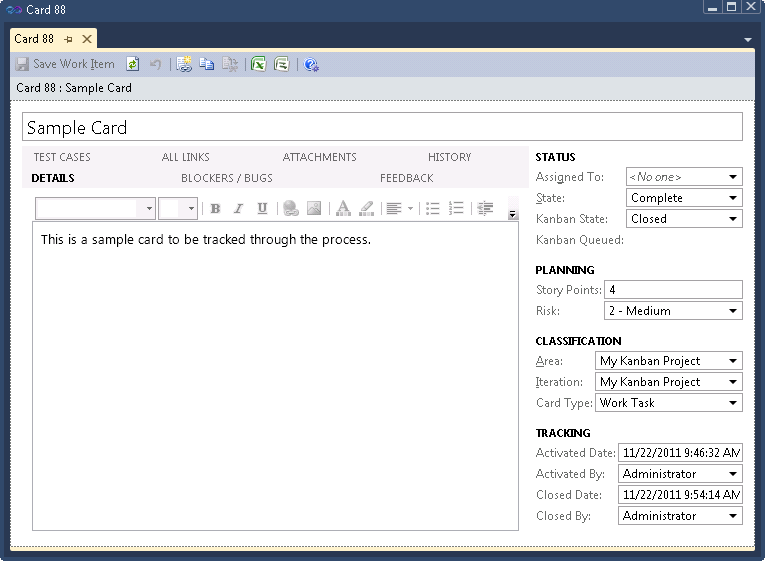


Figure Completing a Card

#### Bugs

Bugs are associated with a Card and must be resolved to allow a Card to move to the next stage. The Bug is a defect in the artifact that is being produced to fulfill the Card.

Bugs follow the normal lifecycle and are raised in the normal ways. They are not described further.

When a Card has open Bugs, the card cannot move to the next Kanban State. While it is in a particular state, it counts towards that state’s WIP limit even if it is blocked.

#### Moving a Card to a Different Stage than the Logical Next One

The Process Template allows you complete flexibility in assigning a Card to a Kanban State. For various reasons you may need to move a Card either back in the process or make it skip a step, you can do this at any time be editing the Kanban State and Kanban Queued fields of the Card so that the Card is in the desired state.

Naturally this can result in a WIP limit being violated. The Process Template does not prevent WIP limits from being violated, however the Kanban Board report allows you to see where WIP limits are being violated.

### **Queries**

The Process Template provides a number of queries to assist you in running a project using Kanban.

#### Planning Queries

This folder contains the following queries:

* All Work Items. This query lists all work items in the project.
* Backlog. This query shows all the Card work items that are in the Planned state. Use this to show see the backlog of work that has not yet been pulled into the pipeline.

#### Process Queries

This folder contains the following queries:

* Process Steps. This query lists the Process Step work items and is used to show the stages in your software development process.

#### Work Queries

This folder contains the following queries:

* Active Problems. This shows the Cards that have the Problem Card Type. Use this to show the failure demand on your team.
* Blockers. This query lists the Cards which have the Blocked field set to “yes”.
* In Progress Cards. This lists all the Cards that are currently being worked on in any of the Kanban states. Use this to see what the team is currently working on.
* My Cards. Lists the Cards currently assigned to the current user. Use this to see what work you are supposed to be doing.
* Queued Cards. Lists all the cards which have completed their current stage and are in the queue for the next stage. When you are ready to pull a new Card into your stage of the process, use this to decide which Card to pull into the stage.

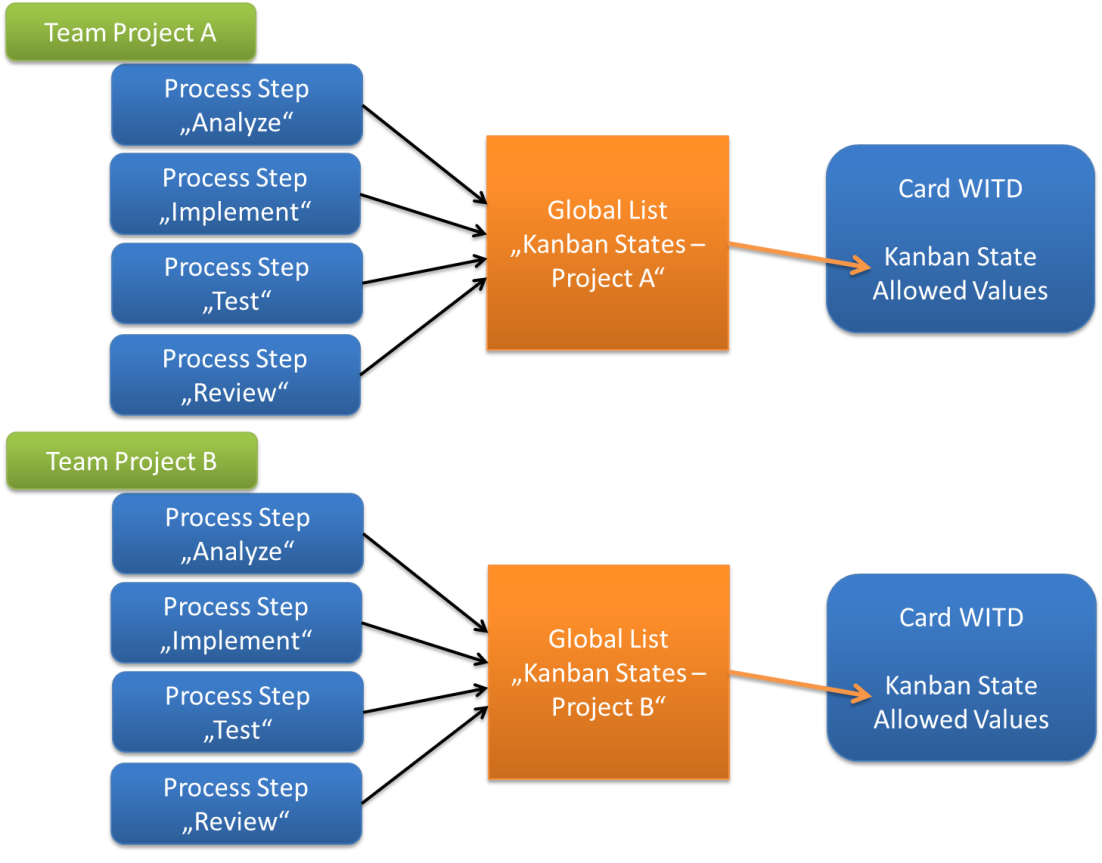
### **Reports**

All the reports use data sourced from the Team Foundation Server Data Warehouse. This provides data that is only a few minutes out of date. For teams that need a real-time Kanban Board, the supported way is to program against the Team Foundation Server object model; however at that point it is recommended that you write an ASP.NET application instead of a report. Such an application is beyond the scope of this guidance.

## GlobalListUpdaterPlugin detail

### Overview

The GlobalListUpdaterPlugin will help you to update the Kanban states on the Card Work Item Type when you change the Process Steps to edit or add Kanban States.

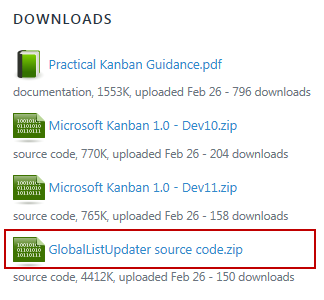
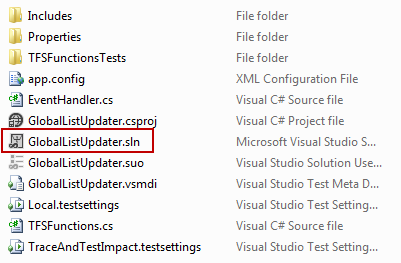
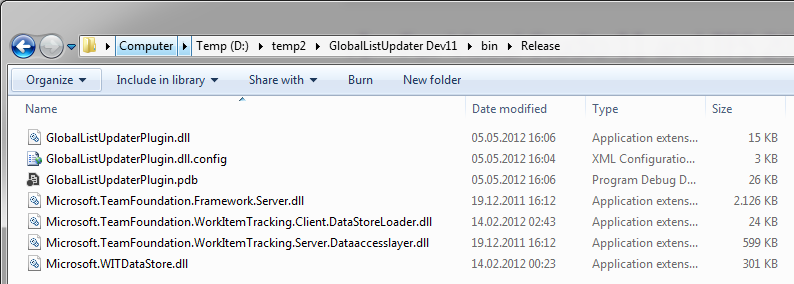


The GlobalListUpdaterPlugin will not only update the content of the Global Lists but also create new ones if they don’t already exist for the corresponding project and then update the Card Work Item Type Definition to use this Global List as allowed values for the Kanban State field. The following picture shows the workflow of the Plugin.



### Setup

The GlobalListUpdaterPlugin is available on CodePlex as source code. You’ll have to compile it yourself before you can use it. Follow the steps below to build and deploy it:

1. Download the GlobalListUpdater source code.zip from <http://vsarkanbanguide.codeplex.com/>   
   
2. Extract the Zip file to a local folder
3. Open the solution file from either the Dev10 or Dev11 folder depending on the version of Team Foundation Server you are using.  
   
4. Make sure the current configuration is set to *Release*
5. Then build the solution by pressing F6
6. You should now see the necessary files in the bin\Release folder of your solutions folder  
   

To use the GlobalListUpdaterPlugin you will have to install it on your Team Foundation Server. Installation is quite simple.

1. Log in to the Application Tier server.
2. For Team Foundation Sever 2010: Copy the compiled assemblies to <System Drive>:\Program Files\Microsoft Team Foundation Server 2010\Application Tier\Web Services\bin\Plugins.
3. For Team Foundation Server 2012: Copy the compiled assemblies to <System Drive>:\Program Files\Microsoft Team Foundation Server 2012\Application Tier\Web Services\bin\Plugins.
4. If you have multiple Application Tiers, repeat step 1-3 for each of them.

The Plugin includes a configuration file which you can use to customize for your specific needs. If you use the default Microsoft Kanban Process Template no changes to the configuration are necessary. See comments for more details on the configuration parameters.

<Rangers.Kanban.GlobalListUpdater.Properties.Settings>

  <!--

  Name of the work item type used to define the Kanban states.

  Default: Process Step)-->

    <setting name="ProcessStepWorkItemType" serializeAs="String">

        <value>Process Step</value>

    </setting>

  <!--

  Prefix for the name of the global lists used for Kanban States.

  The name of the global list is <Prefix> - <Team Project Name>

  Default: Kanban States-->

    <setting name="GlobalListPrefix" serializeAs="String">

        <value>Kanban States</value>

    </setting>

  <!--

  The query executed to read all Process Step work items which will then been

written to the global list.

  @Project and @ProcessStepWorkItemType are valid placeholders-->

    <setting name="ProcessStepsQuery" serializeAs="String">

        <value>SELECT [System.Id], [System.Title] FROM WorkItems WHERE   
[System.TeamProject] = '@project' and [System.WorkItemType] =   
'@ProcessStepWorkItemType' ORDER BY [System.Id]</value>

    </setting>

  <!--

  The Work Items Types whichs definition will be updated if a new global list is

created to use this global list as AllowedValues for the Kanban State.

  You can provide multiple types separated by comma.

  Default: Card-->

    <setting name="WorkItemTypesToChangeAllowedValues" serializeAs="String">

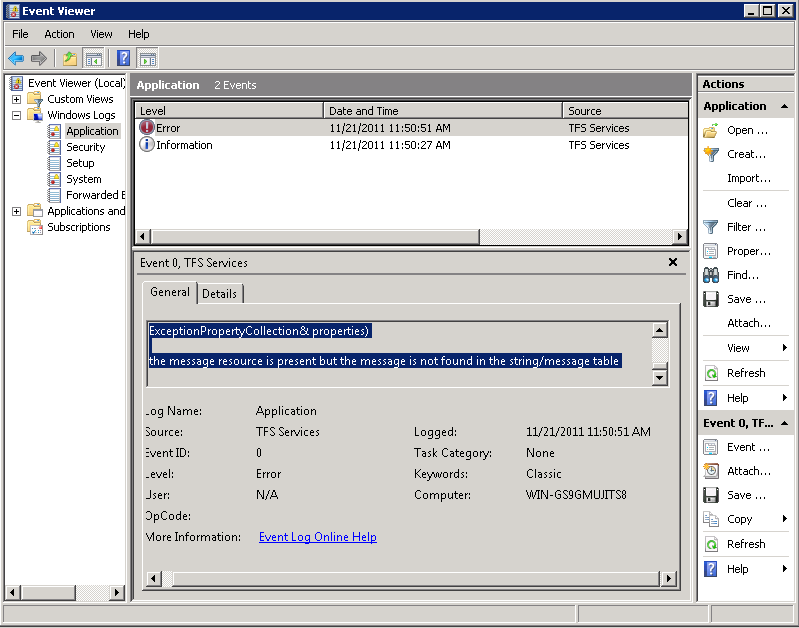
        <value>Card</value>

    </setting>

</Rangers.Kanban.GlobalListUpdater.Properties.Settings>

### Administration

The GlobalListUpdaterPlugin will write messages to the Application Event Log for errors as well as for information. If you update your Process Steps work item and don’t get the expected result, the Event Log is the first place you should check for further information.



## Migrating Existing Projects to the Microsoft Kanban 1.0 Process Template

If you already have an existing Team Project and want to use the Microsoft Kanban 1.0 Process Template with it, there are some steps you have to take to make this work.

There are several ways that this could be done, but the tooling released with this version of the Guidance only supports the method described below.

The supported method of using elements of the Microsoft Kanban 1.0 Process Template is to add the new work item types and reports to the Team Project, and to install the Global List Updater. This requires very little customization, but has the disadvantage that it will not migrate your existing work items to the new structure. Once you have migrated the Team Project, you can start creating new work items based on the Microsoft Kanban 1.0 Process Template. This migration path is most appropriate if you do not have many Work Items in your current backlog.

Future releases of this Guidance and associated tooling may support more complex migration scenarios.

### Adding the New Work Item Types to an Existing Team Project

To add the new work item types to your existing Team Project follow these steps:

1. Download the template from <http://vsarkanbanguide.codeplex.com/> and extract the files to a local directory.
2. Go to the local folder where you have placed the extracted files, and locate the following folder:  
     
   For Team Foundation Server 2010:  
    **Microsoft Kanban 1.0 - Dev10\WorkItem Tracking\TypeDefinitions**For Team Foundation Server 2012:  
    **Microsoft Kanban 1.0 - Dev11\WorkItem Tracking\TypeDefinitions**
3. You’ll see XML files for each work item type in this folder. Most of them are from the MSF for Agile Software Development 6.0 template. You won’t have to update or add these. The two work item types you have to update are the **Process Step** and the **Card**.
4. Start a Visual Studio Command Prompt and change the default directory to the one that contains the work item type definitions above.
5. Use the **witadmin** command to import the Work Item Type Definitions to your existing Team Project

witadmin importwitd /collection:<The URL of your Team Project Collection> /p:<The name of your Team Project> /f:ProcessStep.xml

witadmin importwitd /collection:<The URL of your Team Project Collection> /p:<The name of your Team Project> /f:Card.xml

### Adding the New Queries to an Existing Team Project

You can add the queries which come with the Process Template to your existing project.

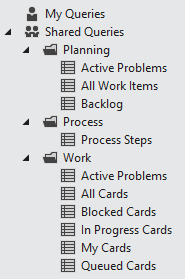


Figure Queries of the Process Template

You can find the Query Definitions in the WorkItem Tracking\Queries folder of the Process Template. The easiest way to import the queries to your project is to download and use the Work Item Query Administration tool from CodePlex (<http://wiqadmin.codeplex.com>). This tool provides a command line command to import query files to a project.

Use this command for each WIQ-File in the template:

wiqadmin import /collection:<Team Project Collection Uri> /p:<Team Project Name> /n:"Shared Queries/<Folder>/<Query Name>" /f:"<Query File>"

The folder can be set equal to the default structure shown in the screenshot above or you can select your own structure. For example the command might look like this:

wiqadmin import /collection:http://vsalm:8080/tfs/test /p:"Kanban 03" /n:"Shared Queries/Work/Active Problems" /f:"Active Problems.wiq"

### Install the Global List Updater

The Global List Updater must be installed by following these steps:

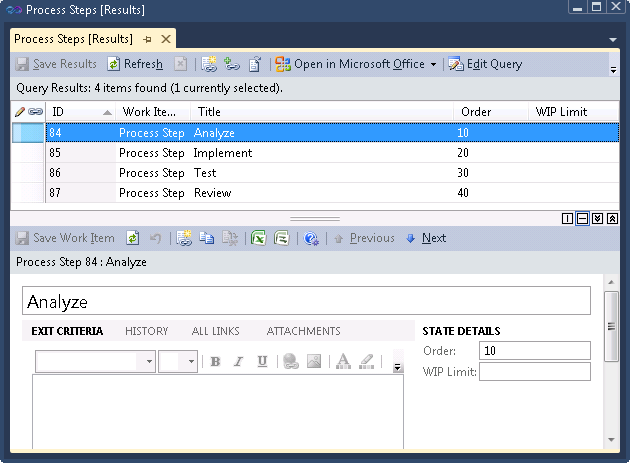
1. Download the GlobalListUpdater source code.zip from <http://vsarkanbanguide.codeplex.com/> and extract the files to a local folder.
2. Compile the GlobalListUpdater and deploy it as described in **Deploying the Global List Synchronizer** on page **19**.
3. Create new Work Items of type Process Step and enter the Title, Order and WIP Limits. You can create the Process Steps according to your needs. If you run a Query selecting all Process Steps, you should get something similar to this:  
     
   

Figure Process Steps

### Adding the New Reports to an Existing Team Project

The next step is to add the reports to the existing Team Project, by following these steps:

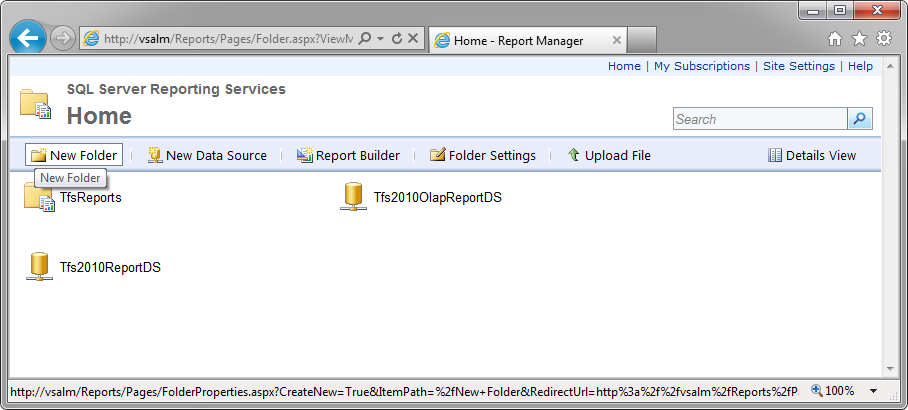
1. Open a Web Browser and browse to http://<Your Reporting Server>/Reports.
2. You should see the Reporting Services Root page, similar to this:  
   

Figure Reporting Services Root Page

1. Navigate to TfsReports/<Your Team Project Collection>/<Your Team Project>/Project Management
2. Click on **Upload File**

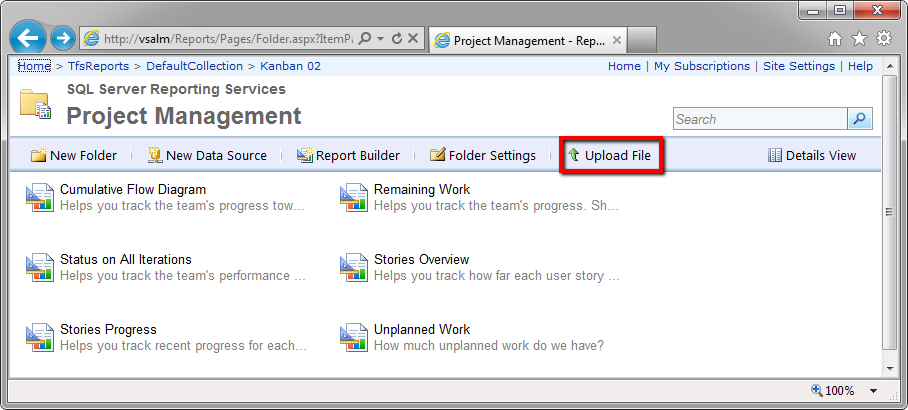


Figure Upload a File

1. Click **Browse** and navigate to the folder you have extracted the Process Template and then to the folder **Reports**
2. Select the Kanban Board.rdl file and upload it.
3. Repeat steps 4-6 for the following files:
   1. CFD.rdl
   2. Kanban Board Sub Report1.rdl
   3. Kanban Board Sub Report2.rdl
   4. Kanban Board Sub Report3.rdl
   5. Kanban Board Sub Report4.rdl
4. Now you should be able to open the Kanban Board report from your browser and see any Cards you have already created.

### Next Steps

You can now start creating Card work items. The Kanban State field should have the Process Steps you defined earlier.

If you want to see existing work items on the kanban board, you can copy them to Card work items as follows:

1. Execute a query in Visual Studio, showing the work items you want to copy
2. Right click on the work item in the Query Results window and select **Create Copy of Work Item**
3. You can now create a copy of the type **Card** as shown below:

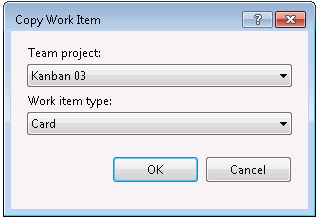


Figure Copy a Work Item

1. Visual Studio will try to map fields with the same name so you should get most of your original work item content on the new Card work item.
2. Repeat steps 2 and 3 for every work item you wish to see on the kanban board.
3. The new Card work items will now show up in the Kanban Board Report.

# Managing and Tracking Progress with Kanban

WHAT’S IN THIS CHAPTER?

* Measurements in Kanban
* Interpreting the Kanban Board report
* Interpreting the Cumulative Flow Diagram report

## Measurements in Kanban

People who use Kanban for software development typically try and reduce their lead time and work in progress. Lead time (LT) is the amount of time that it takes to turn a feature request into delivery. The time typically starts when a customer (or Product Owner) asks for a feature and stops when the feature is delivered to the customer. Stakeholders are able to be more agile and respond more quickly to market changes if teams can deliver high quality features rapidly.

Work in progress (WIP) is the amount of unfinished work that a team has. Kanban teams try and minimize the amount of work in progress by completing tasks before starting new ones.

WIP and LT are related. This relationship is explained by Little’s Law[[1]](#footnote-2). Little’s Law states that there is a proportional relationship between WIP and LT given a constant arrival rate. Translated this tells us that if a team works at a sustained pace, and is fed requirements at the same rate increasing WIP will increase LT. Conversely decreasing WIP will decrease LT. The realization of this theory is one of the principles of Kanban - Minimize *WIP.* The simplest way to reduce lead time is to decrease the amount of work in progress. The Kanban Board report and Cumulative Flow Diagram report can help us achieve this.

## Interpreting the Kanban Board report

The Kanban Board report leftmost column represents un-started work in the backlog and the rightmost column shows tasks that are done or completed. Everything else represents work in progress. Each column or state is typically made up of three things. An in-progress sub-column that shows items currently being worked on, and a Done sub-column that shows items that have completed that state and are ready to be pulled into the next state when there is capacity. Each state also typically has a work in progress or WIP limit. This limit is designed to restrict the number of items that can be in that state. These limits help the team to realize the benefits of lowering lead time described using Little’s Law in the previous section.

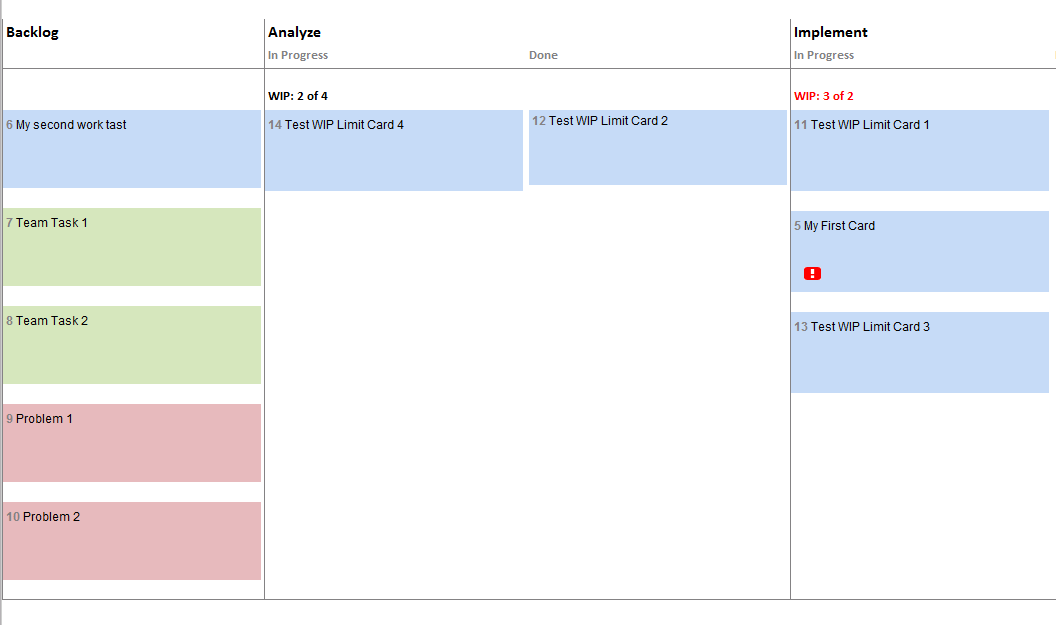


Figure The Kanban Board report

The Kanban Board report shown in Figure 11 The Kanban Board report can be used to track and monitor progress. The Backlog column contains a number of different task types, shown by different colors. The blue tasks represent a work task. The green tasks are team tasks and the red tasks are problems. Clicking the work item id on the report opens the work item as shown in Figure 2: Click through to open the work item.

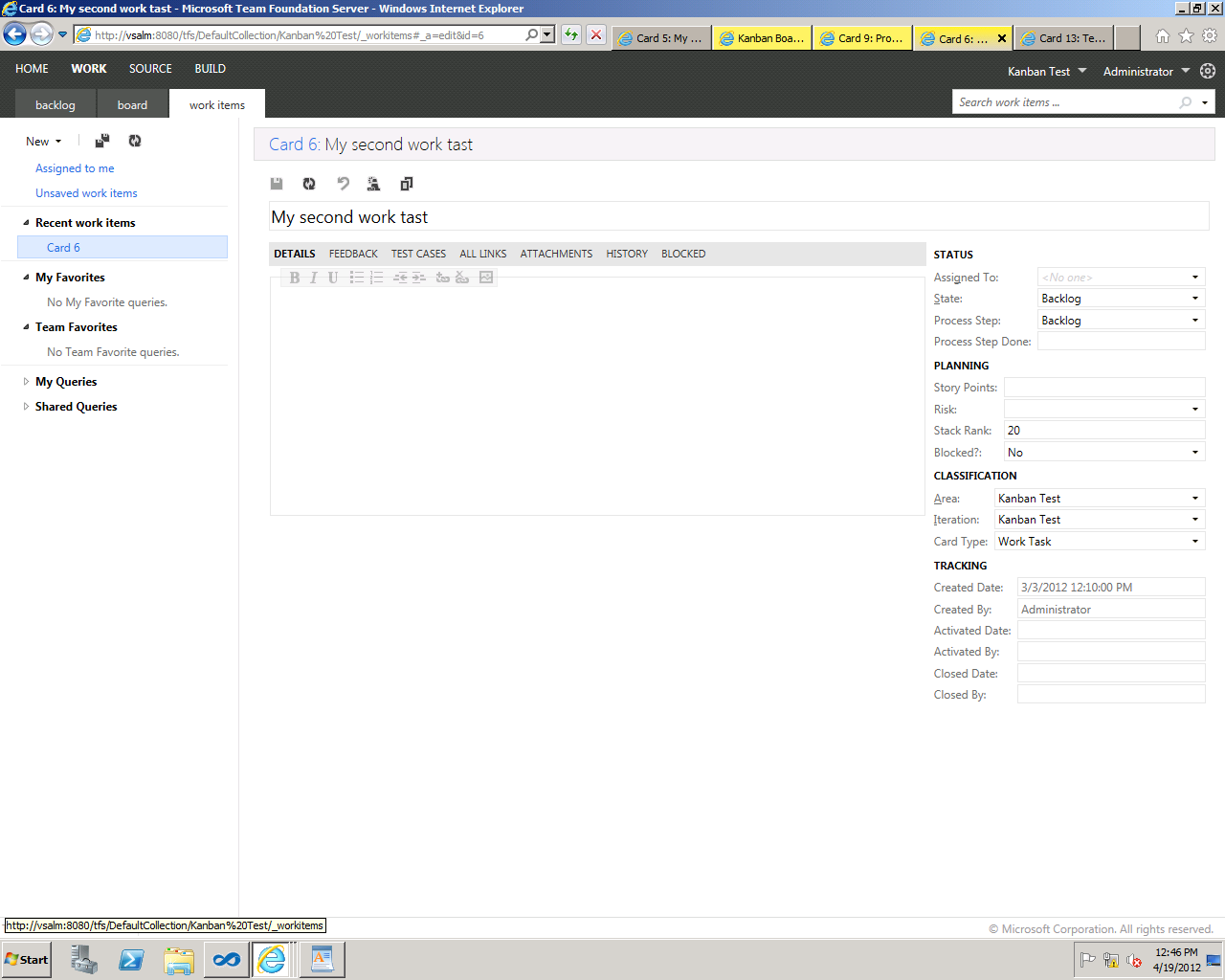


Figure Click through to open the work item

The Analyze column shows one item (12) that is ready to be pulled into the Implement column. The Implement column is in breach of its work in progress limit. This is shown with the red text.

As the Increment column is in breach of its WIP limit, no new Analyze tasks can be started. If nothing is done to rectify this breach then sometimes the upstream columns (in this case Analyze) will breach their limits also. At some point the system becomes clogged and cannot take any more work. The first indicator of this was the breach on the Implement column. This was the warning that the system was having problems; this demonstrates that WIP is a leading indicator of issues.

The final feature shown in Figure 1: The Kanban Board report is linking a bug. The task – “My First Card” shown on the rightmost column has a red exclamation icon. This icon shows that this task has a bug associated with it. Looking across the board for these indicators are a way for the team to see if any quality issues exist with the current work in progress.

## Interpreting the Cumulative Flow Diagram report

Cumulative Flow diagrams (CFD’s) are useful reports to understand and track progress. These reports show the amount of work in certain states over time. By monitoring how much work is in each state we are able to make predictions about lead time and likely completion dates. Figure 13 shows the Cumulative Flow Diagram report.

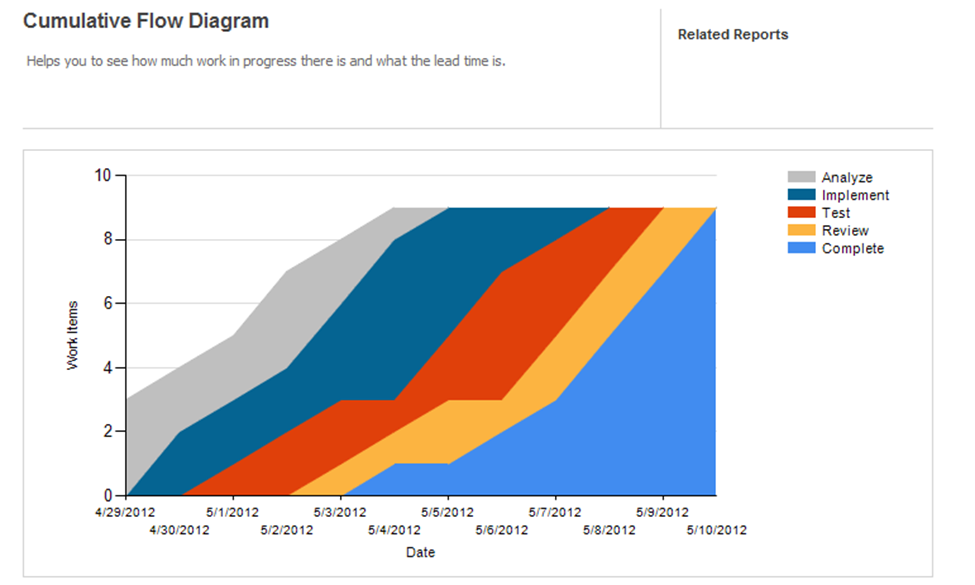


Figure 17 Cumulative Flow Diagram

The CFD is a historical view of the Kanban board. It represents a snapshot of the Kanban board taken each day with counts for how many tasks are in each state.

The horizontal axis of a CFD is shows lead time. The vertical axis shows work in progress. WIP is shown across different states. By monitoring the height (WIP) of the number of tasks in each state we can identify bottlenecks and try and use techniques such as the Theory of Constraints [[2]](#footnote-3)to remove them.

The CFD shows lead time across the horizontal axis and WIP across the vertical axis.

The CFD should be used as a basis for improvement. By monitoring the WIP and lead time shown in the report, the team knows what areas need more work. For example if the WIP and lead time are too high in the Testing state then this is the start point for investigation. Are there enough people with testing skills? Is the test tooling adequate? Are there too many developers? Do the testers have an unwieldy process that can be improved? By using the CFD to highlight areas for improvement, significant gains in lead time can be accomplished.

Figure 14 The Cumulative Flow Diagram showing WIP and Lead Time shows how to measure WIP and lead time.

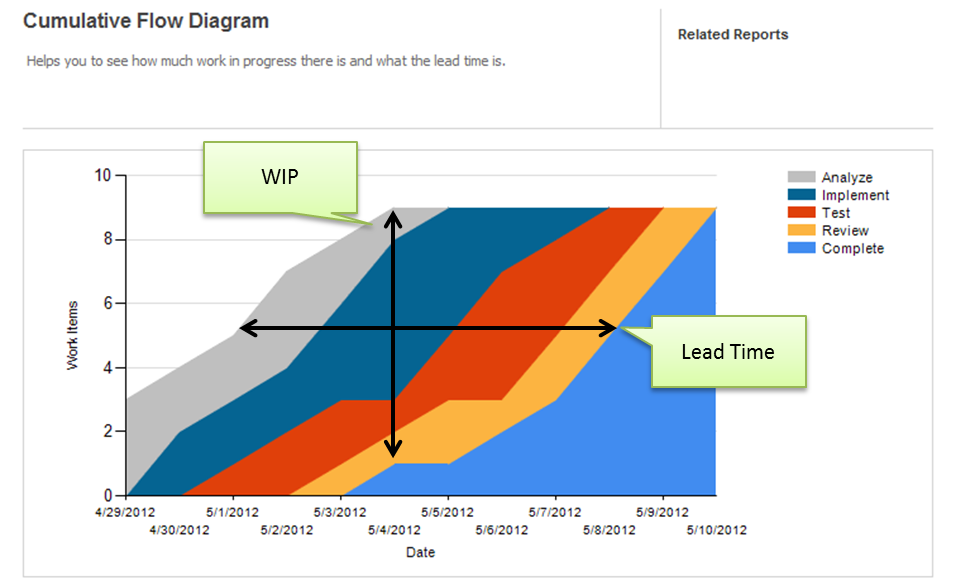


Figure Cumulative Flow Diagram showing WIP and Lead Time

# Further Reading

WHAT’S IN THIS CHAPTER?

* Recommended books.
* Other resources on Kanban.

## Books

*Kanban: Successful Evolutionary Change for Your Technology Business*

David J. Anderson

Blue Hole Press (2010)

If you read one book on Kanban this is the book to read. It provides a thorough practical introduction to Kanban and its principles.

*Kanban and Scrum - making the most of both*

[Henrik Kniberg](http://www.amazon.com/s/ref=ntt_athr_dp_sr_1?_encoding=UTF8&sort=relevancerank&search-alias=books&field-author=Henrik%20Kniberg) , [Mattias Skarin](http://www.amazon.com/s/ref=ntt_athr_dp_sr_2?_encoding=UTF8&sort=relevancerank&search-alias=books&field-author=Mattias%20Skarin)

Lulu.com (2010)

The book describes what Scrum and Kanban have in common and what separates them. Other Kanban books describe what makes Kanban different to Waterfall projects, but this is knowledge that people who already use an Agile process already have. This book is more about helping the reader to decide how to implement an Agile process. It also describes some experiences of introducing Kanban to a team.

*Leading Lean Software Development: Results are Not the Point*

Mary and Tom Poppendieck

Addison Wesley (2009)

This book describes how to implement change within a software development organisation, and make it stick. It focusses on three areas; Systems Thinking, Technical Excellence and Reliable Delivery.

*The Principles of Product Development Flow: Second Generation Lean Product Development*

Don G. Reinertsen

Celeritas Publishing (2009)

This book presents 175 underlying principles and practical methods that help you:

* Improve economic decisions
* Manage queues
* Reduce batch size
* Apply WIP constraints
* Accelerate feedback
* Manage flows in the presence of variability
* Decentralize control

*Lean From The Trenches*

Henrik Kniberg

Unpublished: <http://www.crisp.se/henrik.kniberg/Lean-from-the-trenches.pdf>

This book is a case study of how Kanban was successfully used in a new complex software development project. This book is currently open for pre-order.

*The Goal: A Process of Ongoing Improvement*

Eli Goldratt

Gower Publishing (2004)

This book explains how any process is only as strong as its weakest link. It describes the Theory of Constraints as a method to continually improve a process.

*Freedom from Command and Control: A Better Way to Make the Work Work*

John Seddon

Vanguard Consulting (2003)

This book is about the broader topic of Systems Thinking applied mainly in the context of the customer service industry

*An Introduction to General Systems Thinking*

Gerald M. Weinberg

Dorset House (1975)

A somewhat abstract book that introduces the topic of Systems Thinking, which underpins Kanban. It is interesting to see how that the concept of Systems Thinking is not new.

## Blogs

*David Anderson*

<http://agilemanagement.net/index.php/Blog/>

The blog of the author of the recommended book on Kanban.

*Karl Scotland*

<http://availagility.co.uk/>

*Alan Shalloway*

<http://www.netobjectives.com/blog/9>

*Henrik Kniberg*

<http://blog.crisp.se/author/henrikkniberg>

## Web Sites

*Lean Kanban University*

<http://leankanbanuniversity.com/>

This web site is the *home* of the Kanban Method and has a large amount of training resources

*Kanban 101*

<http://www.kanban101.com/>

This web site gives a high level definition of what Kanban is for software development

*Limited WIP Society*

<http://www.limitedwipsociety.org/>

This web aims to create a central place where its community can gather information together in one place. It also aims to present a consistent message about Kanban to the wider software community. The site gives a high level definition of what Kanban is for software development.

## Other Resources

*Implementing Lean Software Delivery With Kanban and Team Foundation Server 2010.*

<http://channel9.msdn.com/Events/TechDays/TechDays-2011-Belgium/TD035>

Video of Adam Gilmore’s session at Microsoft Tech Days 2011 in Belgium.

*Lean & Kanban 2011*

<http://vimeo.com/channels/leankanban2011benelux>

Videos from the Lean & Kanban 2011 conference in Belgium. Highlights include a critique of Deming, a leading Lean thinker, arguing that Deming’s work applies to manufacturing, but less so to product development, like software development. David Anderson also talks about when Kanban should not be used.

*KanbanDev Yahoo group*

<http://finance.groups.yahoo.com/group/kanbandev/>

This group discusses the use of The Kanban Method to drive change in technology businesses, as defined in David J. Anderson's book, "Kanban - Successful Evolutionary Change for your Technology Business" and Corey Ladas' "Scrumban" and Henrik Kniberg and Mattias Skarin's "Kanban and Scrum - Making the Most of Both."

Implementing Lean Software Delivery with Kanban and Team Foundation Server 2010Implementing Lean Software Delivery with Kanban and Team Foundation Server 2010Implementing Lean Software Delivery with Kanban and Team Foundation Server 2010

1. <http://en.wikipedia.org/wiki/Little%27s_law> [↑](#footnote-ref-2)
2. <http://en.wikipedia.org/wiki/Theory_of_constraints> [↑](#footnote-ref-3)