

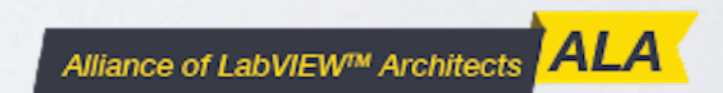
# TIME IS MONEY: LEVERAGING EXTERNAL TOOLS TO STREAMLINE DEVELOPMENT

Darren Mather - iNU Solutions



# A BIT ABOUT ME

- From England, living in Rotterdam
- Certified LabVIEW Architect
- Member of GDevCon
- Founder of the Alliance of LabVIEW Architects
- Giver of Opinions





# CONTENTS

- Introduction
- Background/Discussion
- Useable process flow
- Additional items for more advanced cases
- Using the tools you already have



# INTRODUCTION

- Software developers like to be developing software
- Anything else (other than coffee) is seen as an overhead
- This can lead to huge inefficiencies in your work process
- Could also introduce unnecessary risk



# WHY HAVE A “GOOD” PROCESS

- Firstly, a “good process” is entirely relative
  - A lone developer will need different processes to a large dev team
- Reduce risk associated with delays, budget, rework etc.
- Makes communication easier with stakeholders
- Allows changes to be made in a structured way, reducing impact



# IMPLICATIONS OF A BAD PROCESS

- No SCC means no backup
  - Also makes working in teams very difficult
- No requirements tracking can lead to an incorrect or incomplete solution
  - Makes working in teams impossible
- No testing can result in a program that simply doesn't work
- Lack of structured bug reporting makes releases difficult to manage



# ISSUES CREATING A GOOD PROCESS

- For accurate and useful requirements tracking, you need a good requirements document
- Using SCC effectively requires discipline
- Setting up and executing a good testing plan requires extra time
- Project Manager/Planner needs to have some level of understanding



# THE BIGGEST ISSUE...

- Everything is an indirect cost (an overhead)
  - The time
  - The resources
  - The planning
  - The research



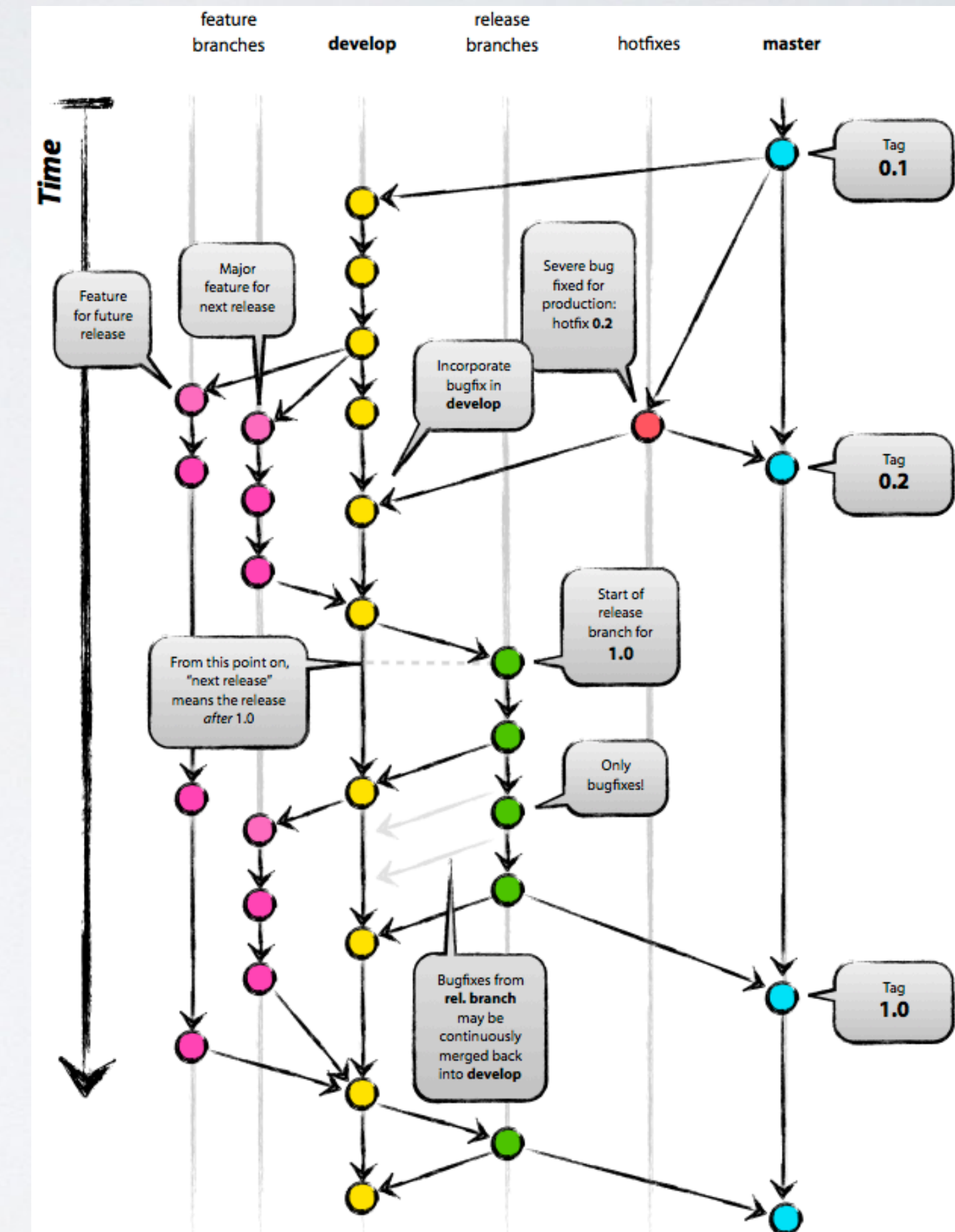
# LEARNING CURVE

- Also Known as a “Panic Graph”
- A lot of the ideas you will already be aware of, but just didn’t know they had technical names
- So the curve may just end up being a gentle bump



# LEARNING CURVE

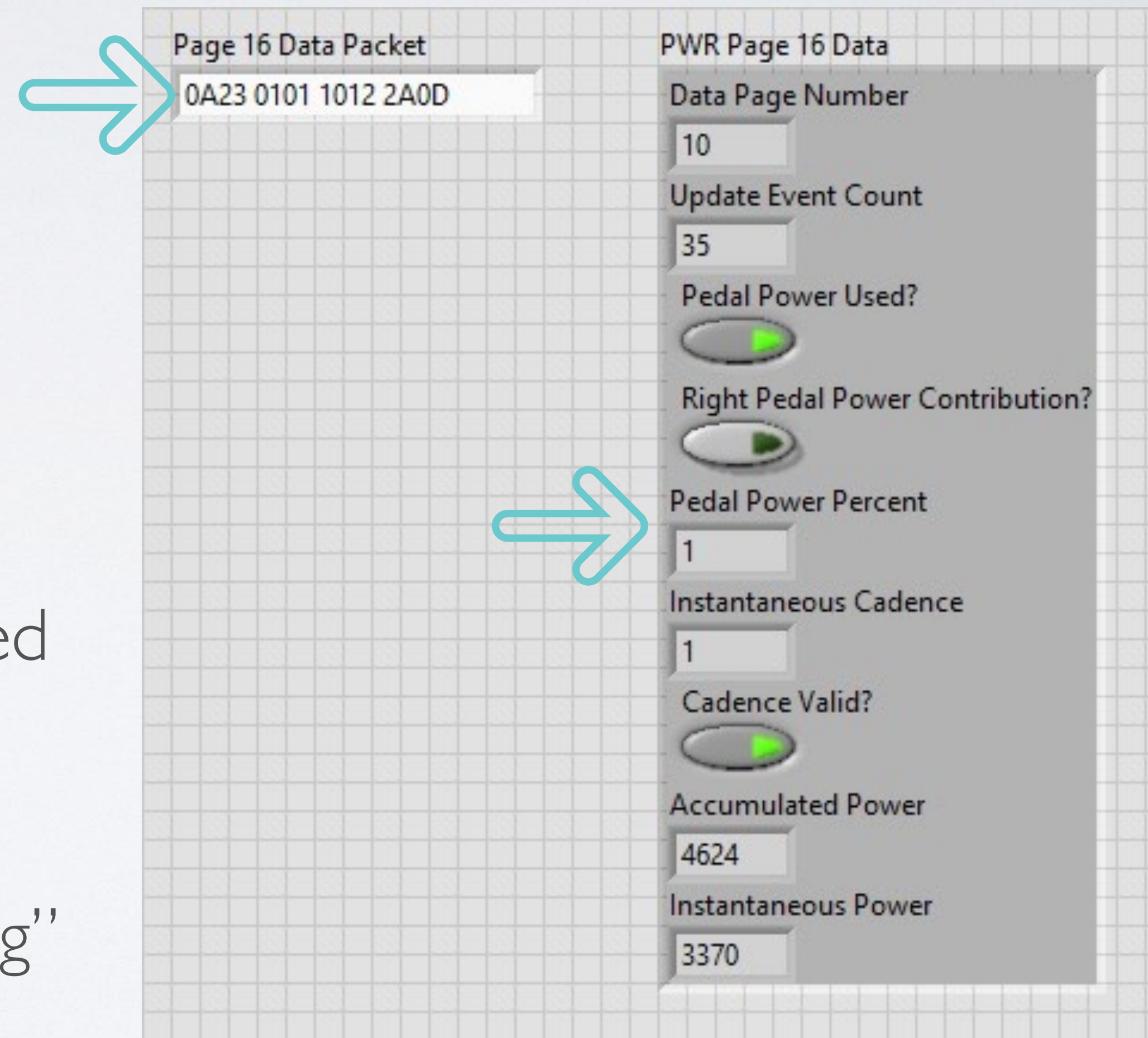
- This is an example of how to use SCC
- I have done it more or less this way for years
- Discovered at CLA Summit 2018 that it is called “Hg-flow”





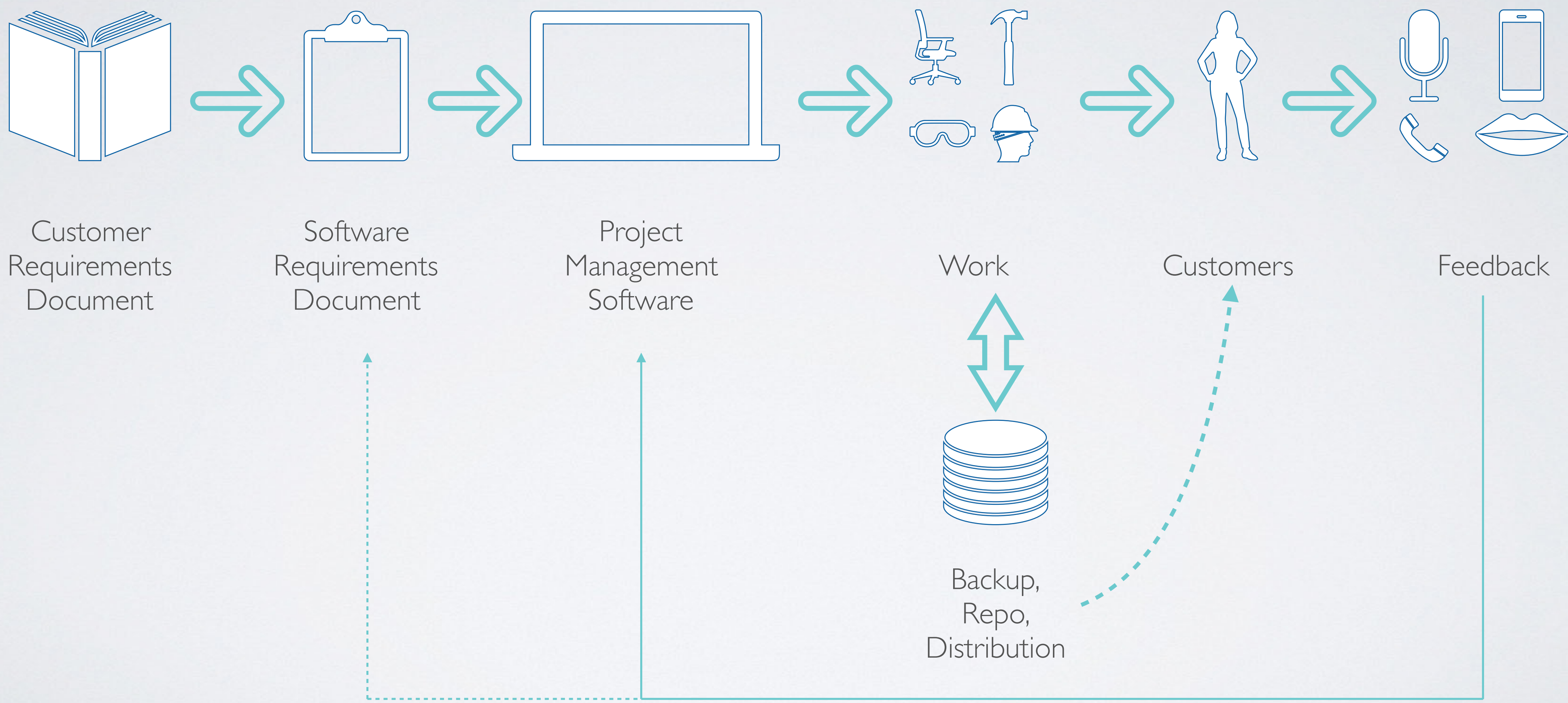
# LEARNING CURVE

- Testing a SubVI
- Put data manually into the controls
- Run it
- See if the indicators show the expected results
- Turns out that this is called “unit testing”





# A USEABLE PROCESS





# REQUIREMENTS DOCUMENT(S)

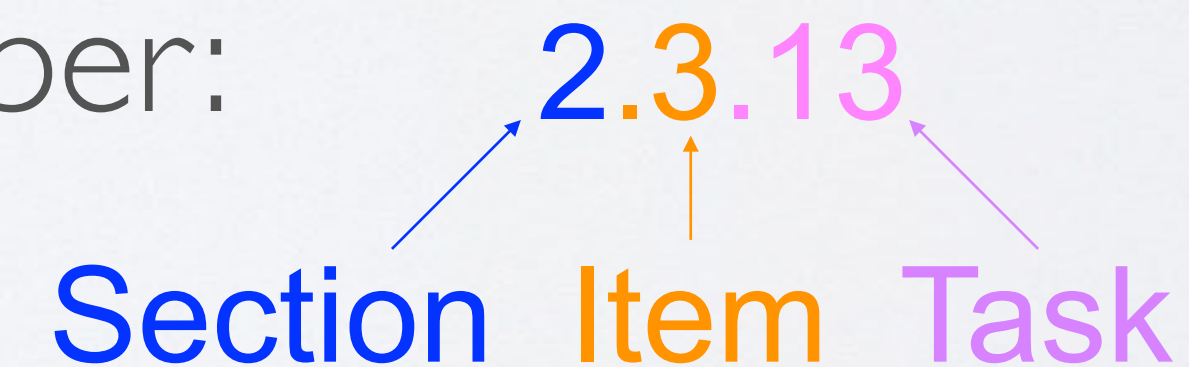
- To have requirements tracking, you need to have requirements
- Most projects will have some kind of document explaining what it is they want to achieve
- You need to translate the “project” requirements into the “software” requirements
- But be kind to yourself, think about the next step in the process...



# CREATING THE SOFTWARE REQUIREMENTS DOCUMENT

- Identify which bits are relevant to you
- Break down the information into logical sections
- Give each Section/Item/Task a number:

2.3.13  
Section Item Task





# CREATING THE SOFTWARE REQUIREMENTS DOCUMENT

## 3. Vacuum Control

### 3.1 Automatic Controls

#### 3.1.1 Pressure Gauge auto on

#### 3.1.2 Warm up time on gauge

#### 3.1.3 Stability monitoring

### 3.2 Manual Controls

#### 3.2.1 Re-pressurising the Turbo Pump

#### 3.2.2 Opening of valves for venting

#### Vacuum control

- Automatic control for:
  - o Switching on/off pressure gauges:
    - During nominal operation (as stated in the calibration sequence)

(\*) note that after switching on a gauge a minimum warming up time of 3 hours should be taken to ensure a proper measurement, before starting a calibration

(\*\*) This is standard for the wst gauges. However as we could have a wide selection of gauges to be calibrated, it would not be possible to account for this in the programming. A power group should be made for both high and low "customer" gauges, which can be programmed to switch the power on/off.

- o Pressure (stability) control of the VGCS (using the 3 wst as input)
- o Starting of degassing of the wst ion gauge.
- o Switching off the TMP when the maximum operating pressure has been reached (1-10<sup>-2</sup>mbar).
- o Re-pressurising the TMP (to be done by using the TMP EXT75DX, controller).

#### N<sub>2</sub> pressure control and facility temperature monitoring

- Controlling and monitoring the N<sub>2</sub> supply pressure from the laboratory
- Monitoring the temperature of the facility (temperature measurement will be supplied from the data acquisition)

#### Motion control

- Closing of the gate valves between the pumping systems and the vessel when the pressure reaches 1-10<sup>-2</sup>mbar (Pneumatic gate valves need to be purchased by customer)

All actions should be able to be performed automated and manually.

#### Vacuum control

- Automatic control for:
  - o Switching on/off pressure gauges:
    - During nominal operation (as stated in the calibration sequence)

(\*) note that after switching on a gauge a minimum warming up time of 3 hours should be taken to ensure a proper measurement, before starting a calibration

(\*\*) This is standard for the wst gauges. However as we could have a wide selection of gauges to be calibrated, it would not be possible to account for this in the programming. A power group should be made for both high and low "customer" gauges, which can be programmed to switch the power on/off.

- o Pressure (stability) control of the VGCS (using the 3 wst as input).
- o Starting of degassing of the wst ion gauge.
- o Switching off the TMP when the maximum operating pressure has been reached (1-10<sup>-2</sup>mbar).
- o Re-pressurising the TMP (to be done by using the TMP EXT75DX, controller).

#### N<sub>2</sub> pressure control and facility temperature monitoring

- Controlling and monitoring the N<sub>2</sub> supply pressure from the laboratory
- Monitoring the temperature of the facility (temperature measurement will be supplied from the data acquisition)

#### Motion control

- Closing of the gate valves between the pumping systems and the vessel when the pressure reaches 1-10<sup>-2</sup>mbar (Pneumatic gate valves need to be purchased by customer)

All actions should be able to be performed automated and manually.

No need to print this out  
can do it in Adobe Acrobat

Task  
Information  
Important



# REQUIREMENTS TRACKING - WHY

- Requirements document becomes the reference for the whole project
- Can be used to create individual tasks, which are then assigned to devs
- Necessary when working in teams, especially multi location teams
- Can also be used to create Gantt charts, auto progress reports etc.
- Creates a link between tasks, progress and milestones/deadlines



# REQUIREMENTS TRACKING - HOW

- Many online/cloud based services (SaaS)
- Most common are products such as JIRA, TFS and OpenProject
- Create a project, add tasks/issues, create sprints and so on...
- Ensure everyone has access, this shouldn't be privileged information



# REQUIREMENTS TRACKING - HOW

- Use your Requirements Document to create the main “topics”
  - Epics
  - User Stories
  - Sprints
- This is where using a “standard” structure for you req. spec. comes in handy



# REQUIREMENTS TRACKING - HOW

- The “section” can become the Epic → **3. Vacuum Control**
- The “item” can become the Stories → **3.1 Automatic Controls**
  - 3.1.1 Pressure Gauge auto on**
  - 3.1.2 Warm up time on gauge**
  - 3.1.3 Stability monitoring**
- The “task” can become the Tasks and/or Sub Tasks → **3.2 Manual Controls**
  - 3.2.1 Re-pressurising the Turbo Pump**
  - 3.2.2 Opening of valves for venting**
- This can be adapted to your preference



# REQUIREMENTS TRACKING - HOW

- When you have all the “tasks” you can then create the Sprints
- Could also be called the Work Packets
- Each task is assigned to a dev/dev team
- All tasks will have a due date and/or expected duration specified



# REQUIREMENTS TRACKING - HOW

The screenshot shows the Jira Software interface with a backlog of issues. Annotations point to various fields and elements:

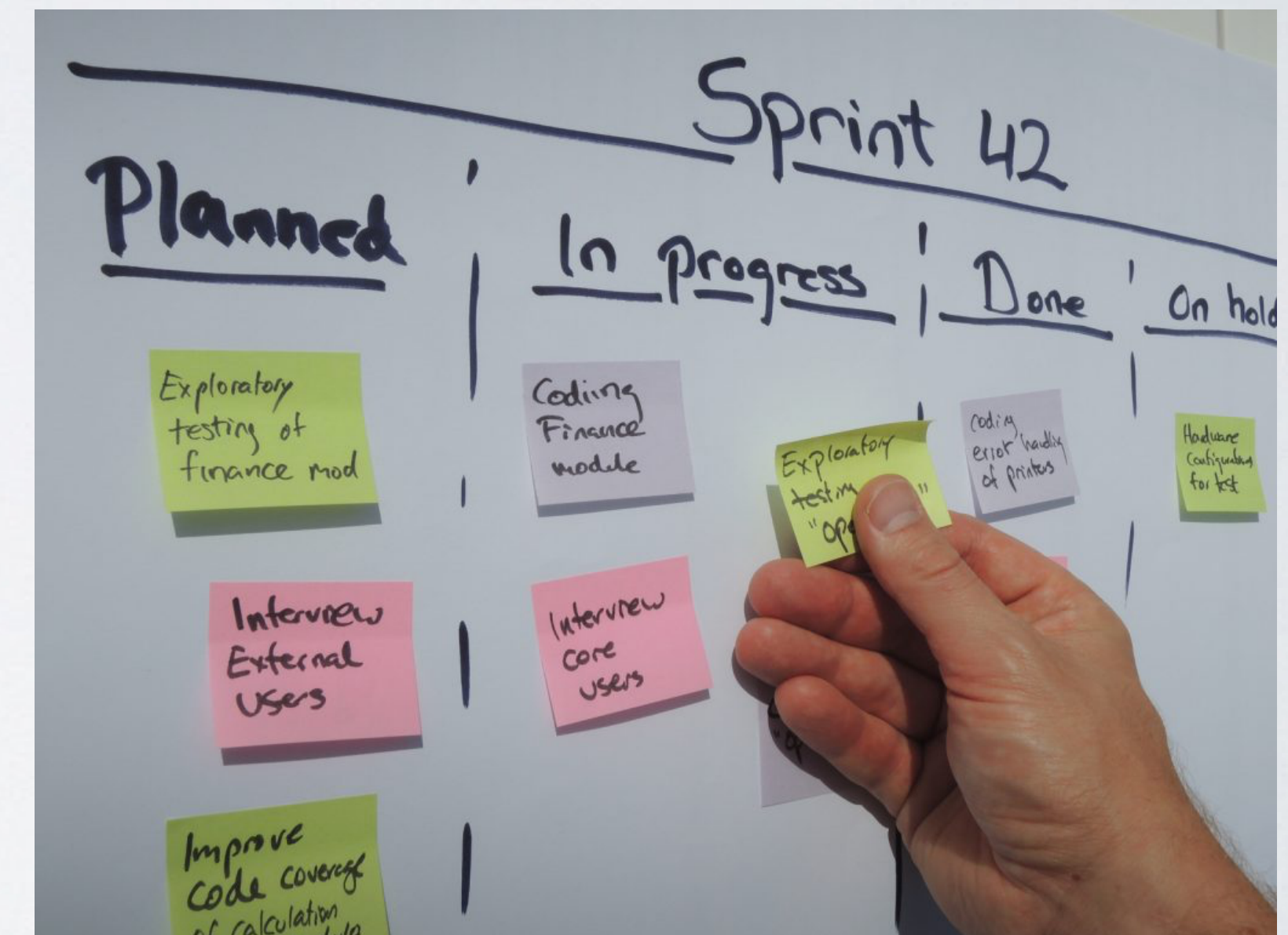
- Story Name:** Points to the issue title "3.1 Automatic Controls".
- Task Name:** Points to the issue title "3.2.1 Repressurising the Turbo Pump".
- Epic Name:** Points to the issue title "3.1.1 Pressure Gauge Auto On".
- Labels:** Points to the "Automatic" and "Vacuum" labels.
- Time Estimate:** Points to the "None" time estimate.
- Sprint:** Points to the "NIW Sprint 2" sprint.
- Description:** Points to the description text: "Implement routine to automaticallt turn on pressure gauges when vacuum in chamber reaches levels of operation for the valve. Needs to work fo all valves, so has to have selector for valve type and minimun pressure constant input".
- Relationship:** Points to the "Linked issues" section showing a relationship to "NIW-6 3.1 Automatic Controls".

The Jira interface includes a top navigation bar with "Jira Software", "Dashboards", "Projects", "Issues", "Boards", and "Create". The left sidebar shows "VERSIONS" and "EPICS". The main area displays a "Backlog" with "NIW Sprint 1" and "NIW Sprint 2". The right sidebar shows details for "NIW-8 3.1.1 Pressure Gauge Auto On", including status, assignee, labels, time estimate, epic link, sprint, priority, reporter, and description.



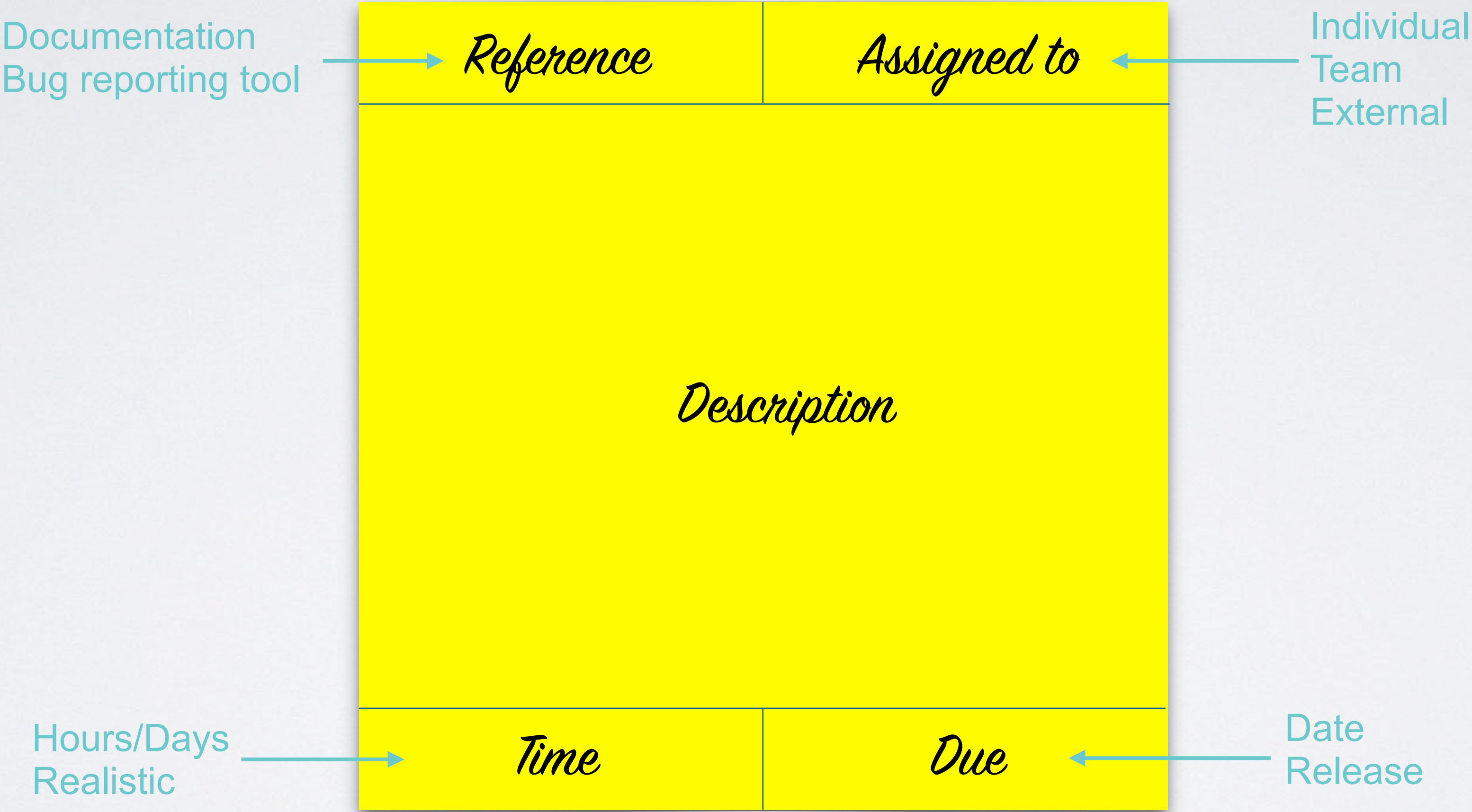
# REQUIREMENTS TRACKING - HOW

- If you prefer something a bit more old school
- Use different colours for Bugs, Issues, Tasks etc.
- Can be the focus of a morning “Stand Up” meeting
- Can create a template for each item





# REQUIREMENTS TRACKING - HOW





# REQUIREMENTS TRACKING - BENEFITS

- Can link all work directly into your Lean/Agile processes
- Can create a daily overview of progress
- Total control over task allocation
- Reduced risk of duplication
- Allows multi-location teams to collaborate effectively
- Can identify dependencies and plan accordingly



# REQUIREMENTS TRACKING - USING

- During development you will need to keep track of completed tasks
  - In your planning software
  - In your code
- Those who have done the CLA exam will remember the Requirements Gateway
- There are several (better) ways to do this...

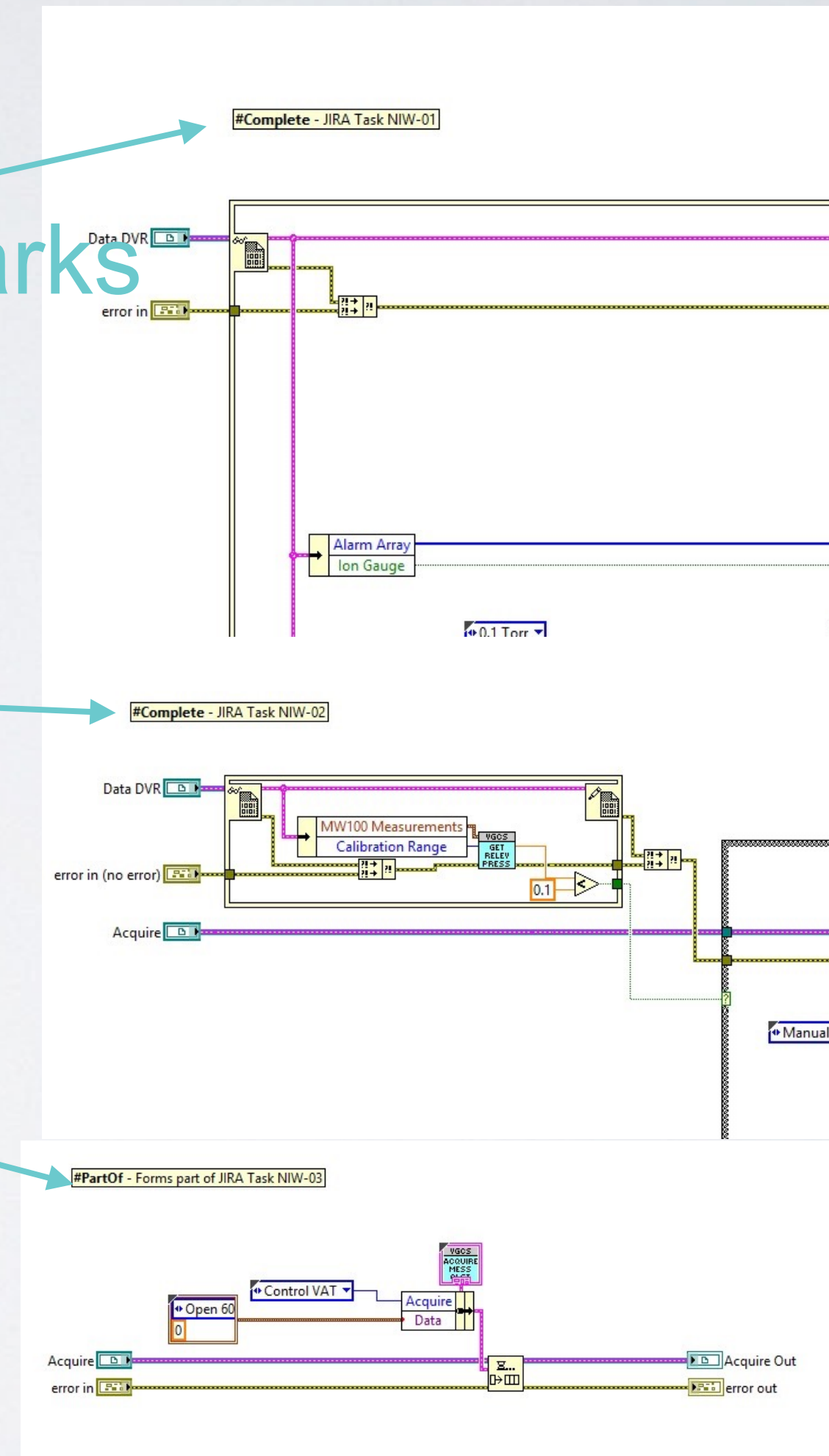
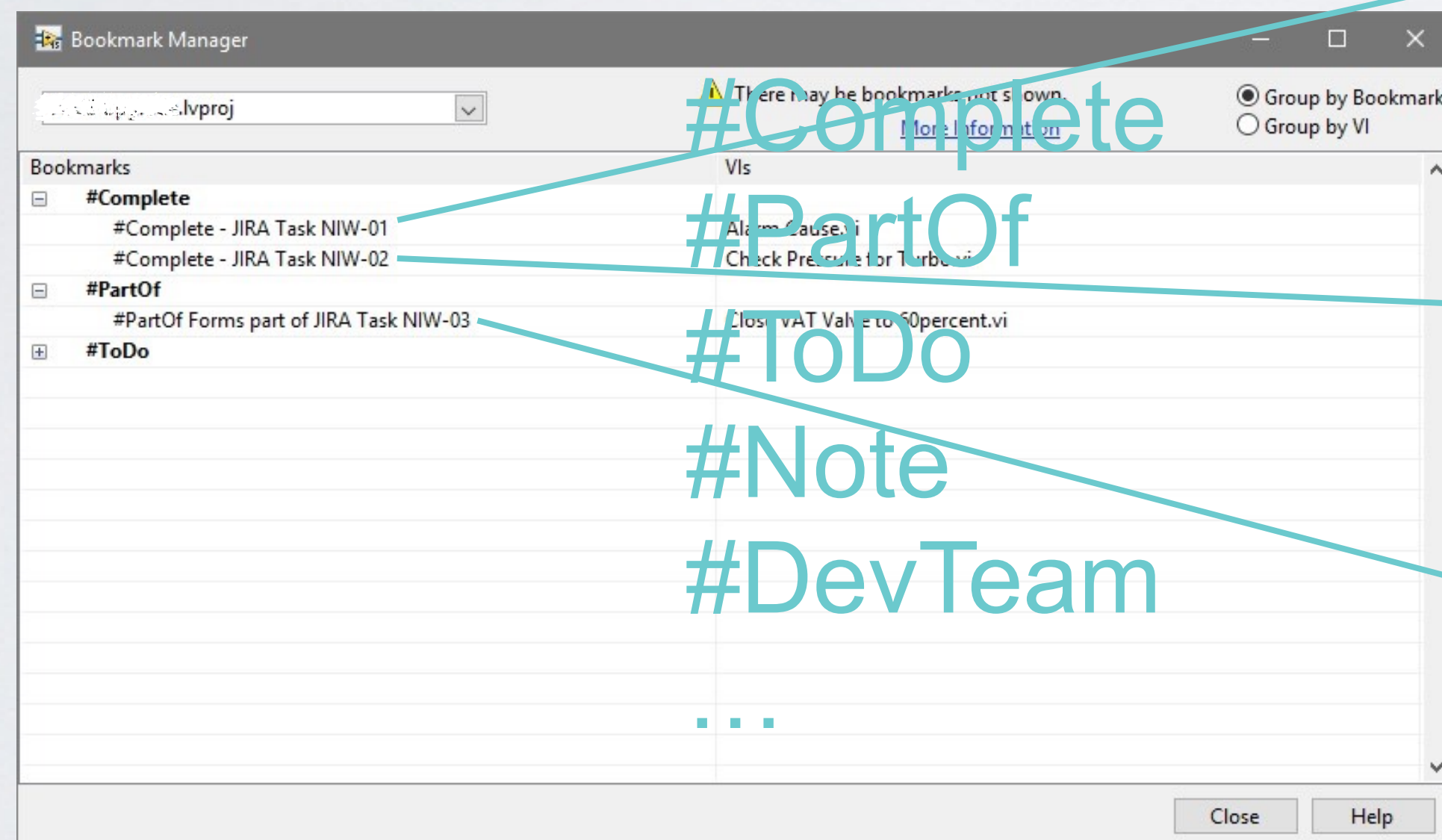


# REQUIREMENTS TRACKING - USING

On the Block Diagram

Use Bookmark Manager

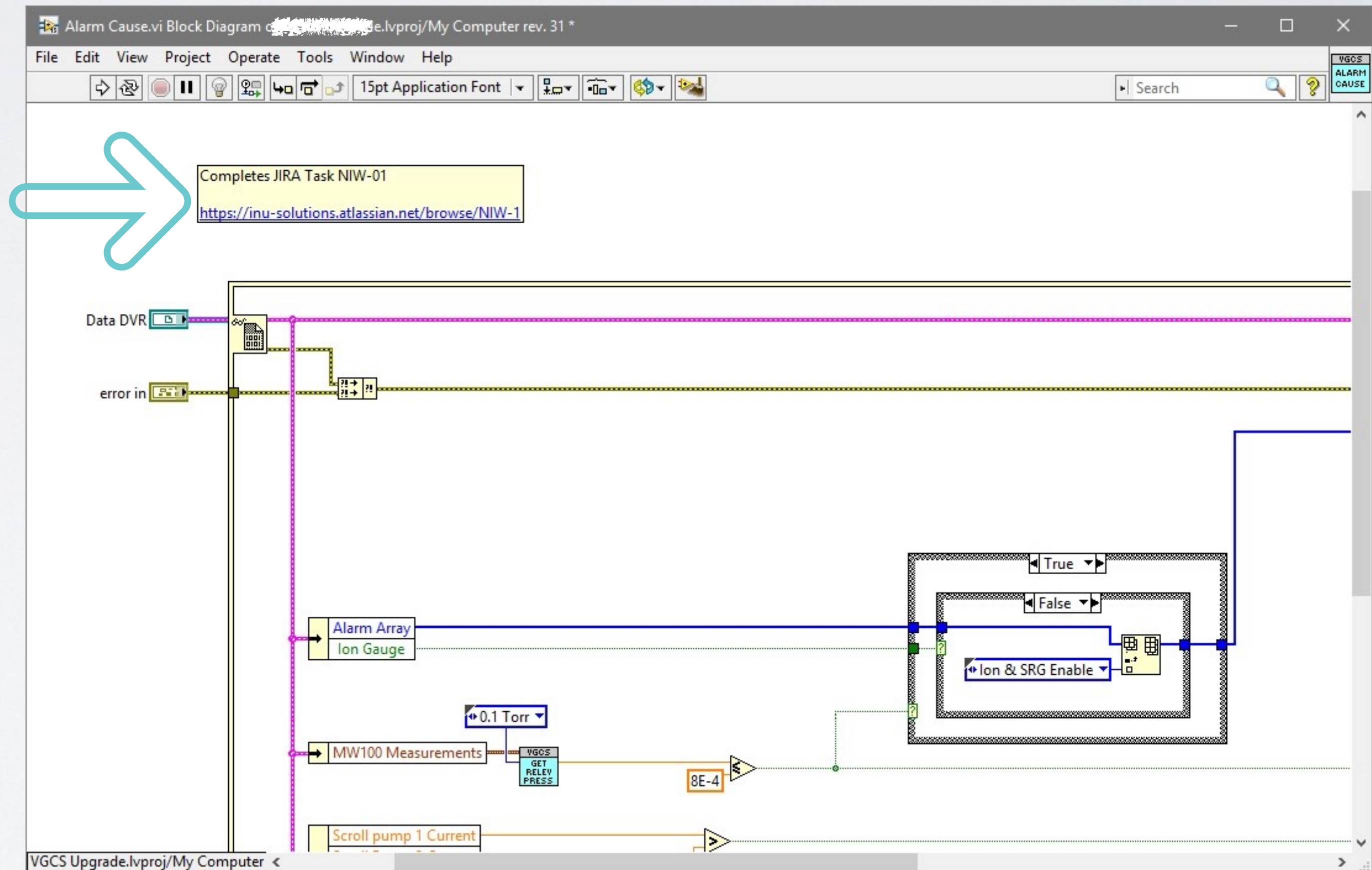
Tip: Create a standard list of BookMarks





# REQUIREMENTS TRACKING - USING

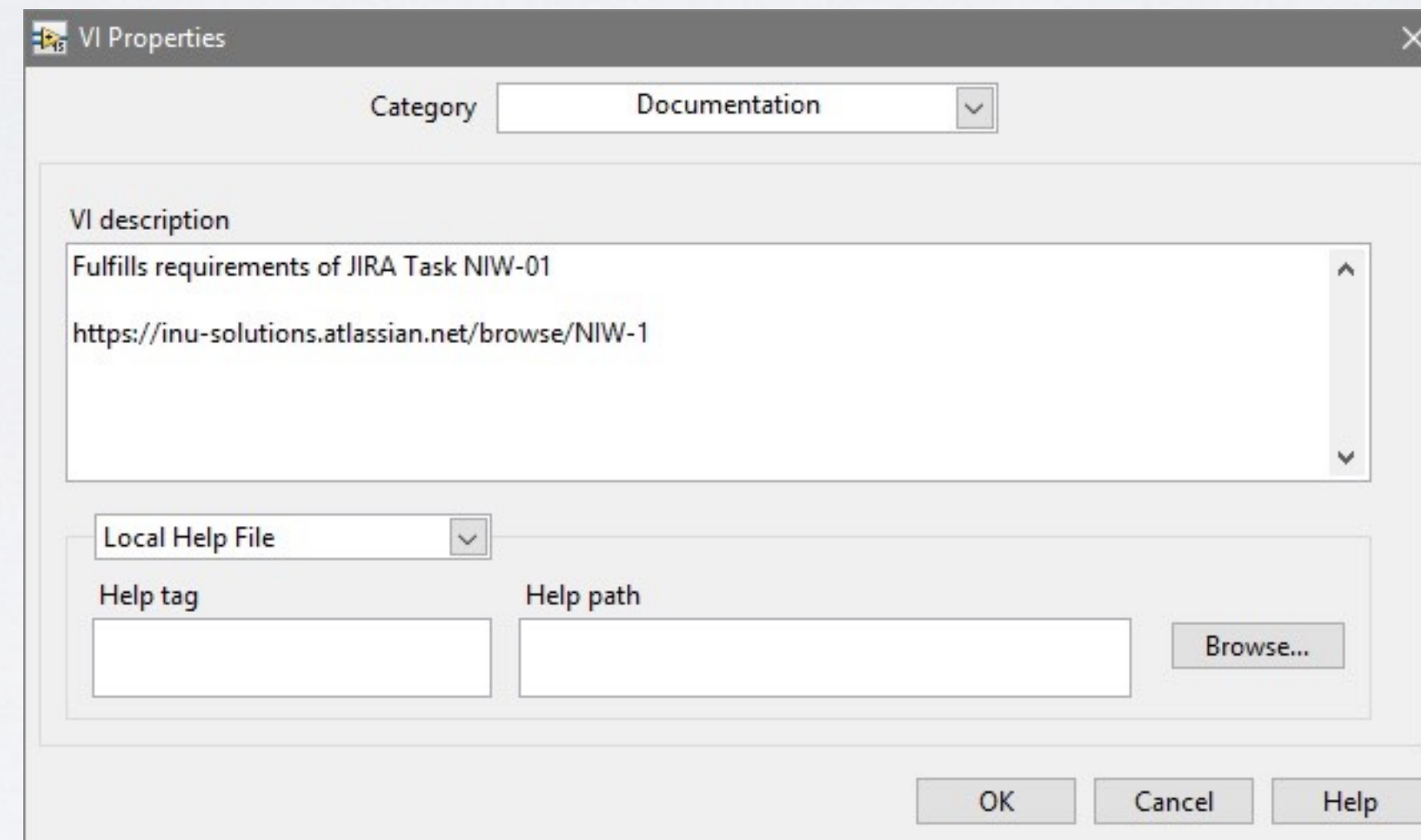
On the Block Diagram  
Add Link to Task





# REQUIREMENTS TRACKING - USING

On the Block Diagram  
VI Description



The screenshot shows a 'VI Properties' dialog box with the 'Documentation' category selected. The 'VI description' text area contains the text 'Fulfills requirements of JIRA Task NIW-01' and a URL 'https://inu-solutions.atlassian.net/browse/NIW-1'. Below this, there is a 'Local Help File' dropdown menu, and two input fields for 'Help tag' and 'Help path', with a 'Browse...' button next to the 'Help path' field. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Category: Documentation

VI description

Fulfills requirements of JIRA Task NIW-01

<https://inu-solutions.atlassian.net/browse/NIW-1>

Local Help File

Help tag

Help path

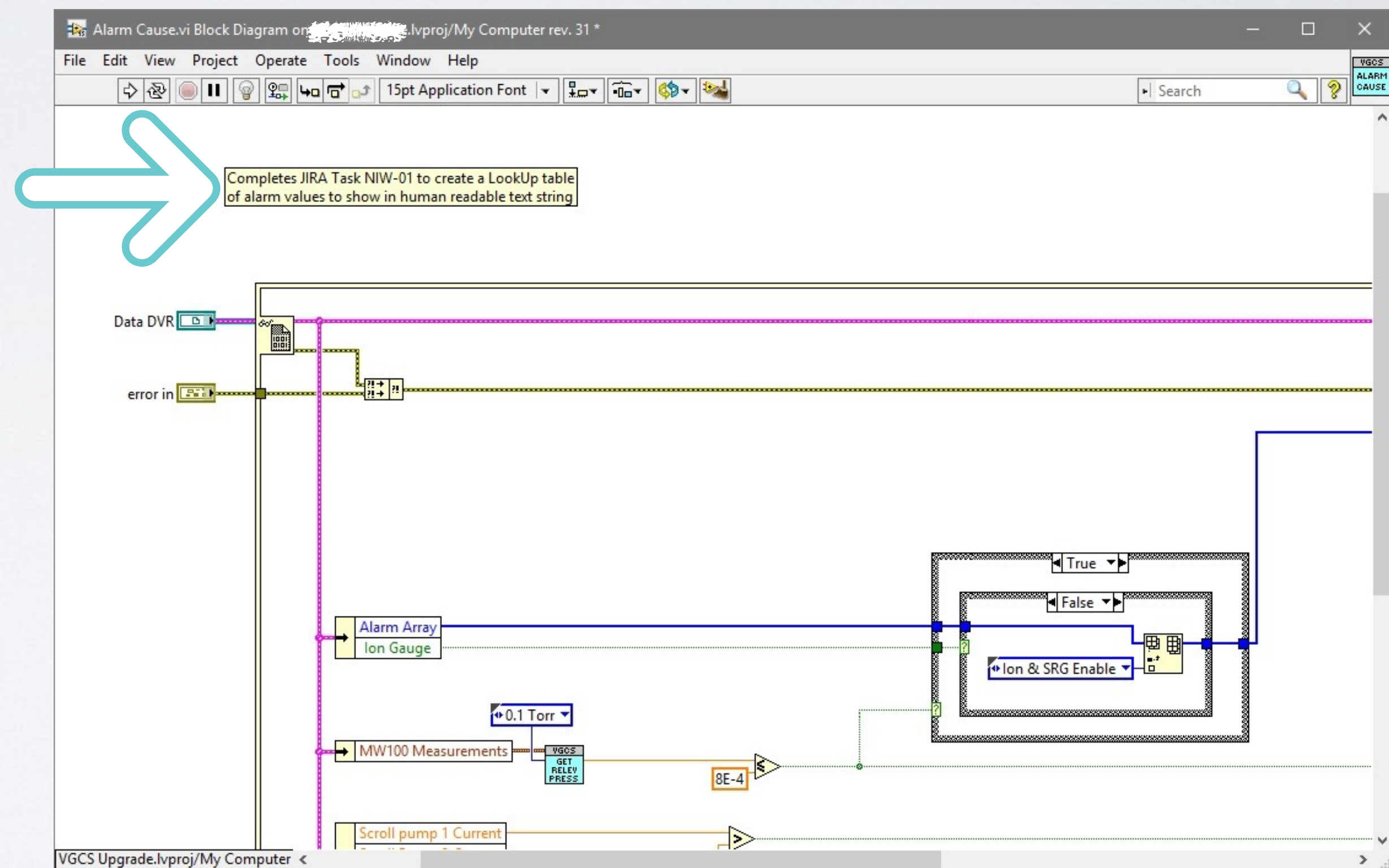
Browse...

OK Cancel Help



# REQUIREMENTS TRACKING - USING

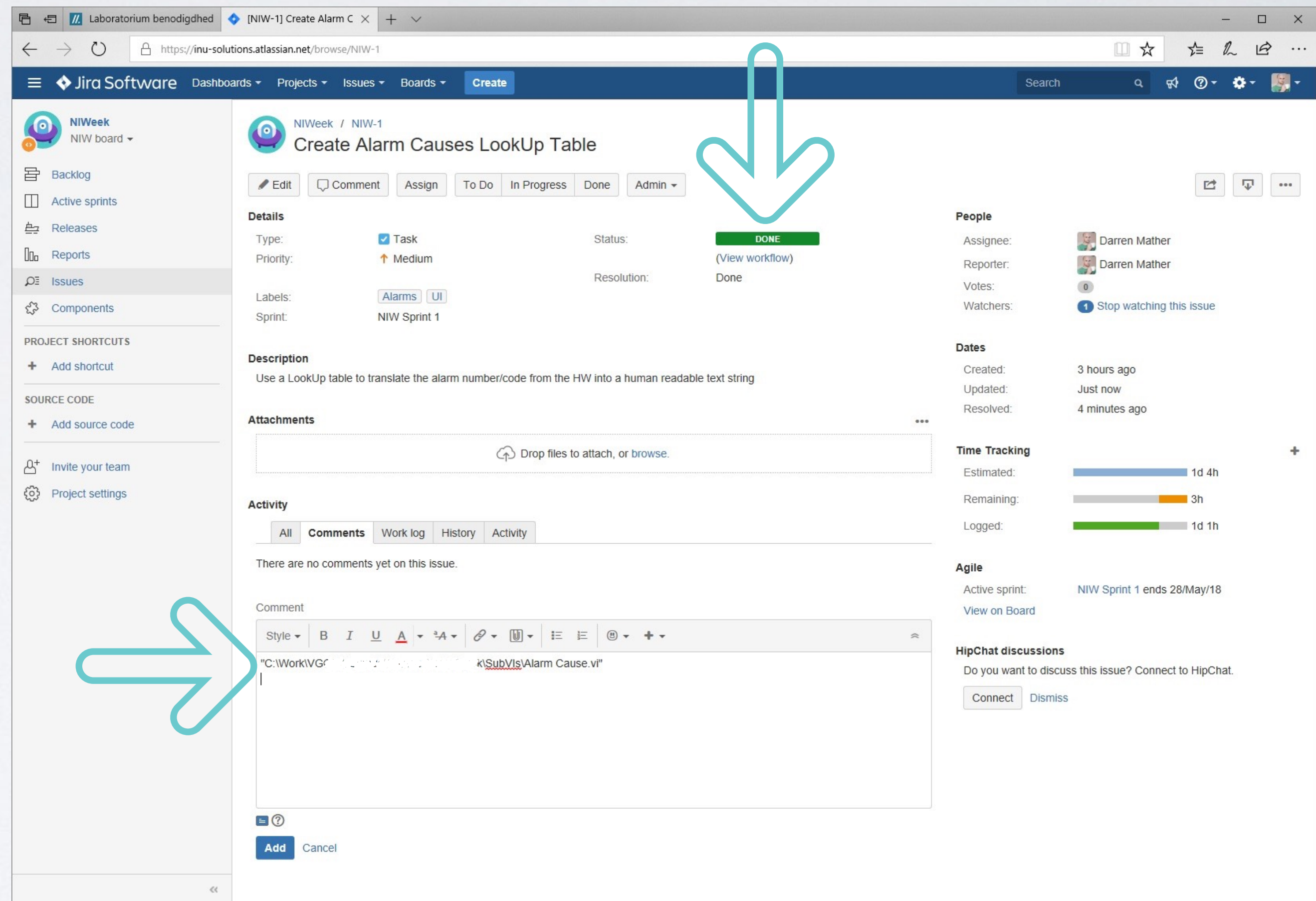
On the Block Diagram  
Make a Simple Note





# REQUIREMENTS TRACKING - USING

In Management Software  
Link to the specific VI

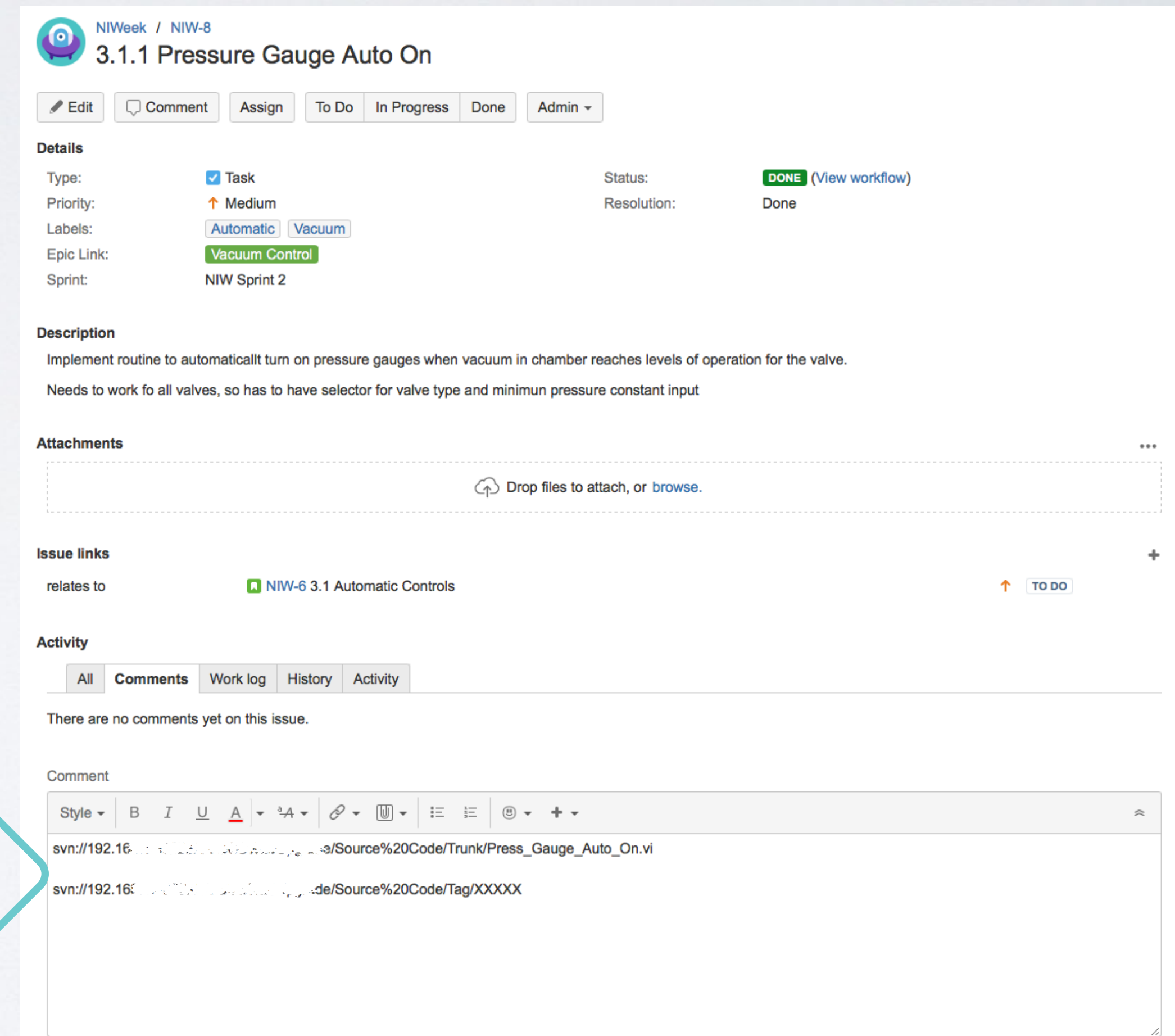


The screenshot displays the Jira Software interface for a project named 'NIWeek'. The main view shows an issue titled 'Create Alarm Causes LookUp Table' with a status of 'DONE'. The issue details include a type of 'Task', a priority of 'Medium', and labels 'Alarms' and 'UI'. The description states: 'Use a LookUp table to translate the alarm number/code from the HW into a human readable text string'. The 'Attachments' section is empty, and the 'Activity' section shows no comments. A large blue arrow points to the 'DONE' status button. Another large blue arrow points to the 'Add' button in the comment section, which is currently empty.



# REQUIREMENTS TRACKING - USING

In Management Software  
Link to Tag/Revision in SCC

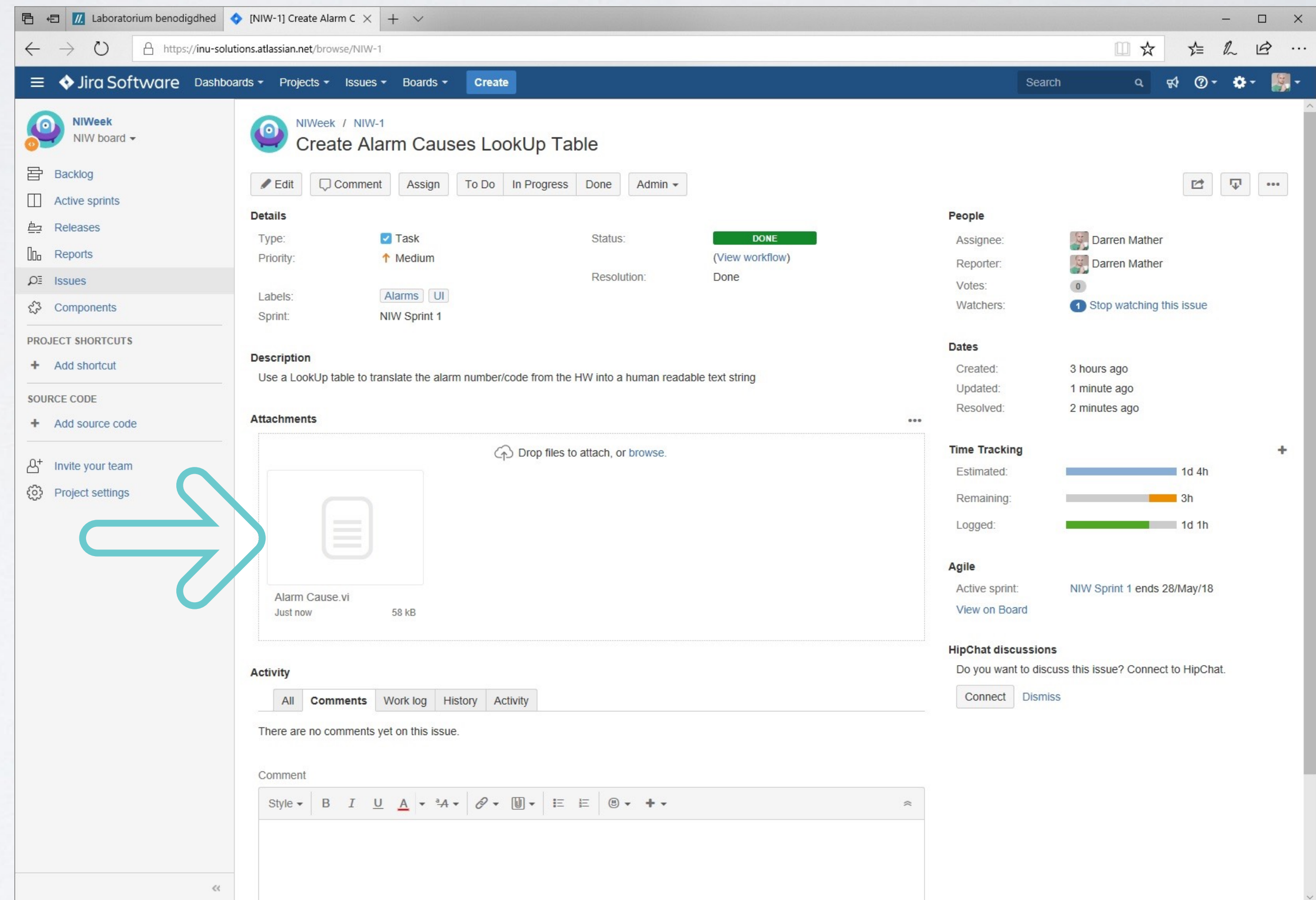


The screenshot shows a Jira issue page for '3.1.1 Pressure Gauge Auto On' under the 'NIWeek / NIW-8' project. The issue is in the 'Done' status. The 'Details' section shows it is a 'Task' with 'Medium' priority, labeled 'Automatic' and 'Vacuum', linked to 'Vacuum Control' and 'NIW Sprint 2'. The 'Description' states: 'Implement routine to automaticallt turn on pressure gauges when vacuum in chamber reaches levels of operation for the valve. Needs to work fo all valves, so has to have selector for valve type and minimun pressure constant input'. The 'Attachments' section is empty. The 'Issue links' section shows it 'relates to' 'NIW-6 3.1 Automatic Controls'. The 'Activity' section shows no comments. A large blue arrow points to the 'Comment' field, which contains two lines of text: 'svn://192.168.1.100:3699/Source%20Code/Trunk/Press\_Gauge\_Auto\_On.vi' and 'svn://192.168.1.100:3699/Source%20Code/Tag/XXXXX'.



# REQUIREMENTS TRACKING - USING

In Management Software  
Attach VI to Task (probably not dynamic)



The screenshot shows a Jira Software interface for a project named 'NIWeek'. The task 'Create Alarm Causes LookUp Table' is displayed, with a status of 'DONE'. The task details include a type of 'Task', a priority of 'Medium', and labels 'Alarms' and 'UI'. The description states: 'Use a LookUp table to translate the alarm number/code from the HW into a human readable text string'. A large blue arrow points from the left sidebar to the 'Attachments' section, where a file named 'Alarm Cause.vi' (58 kB) is attached. The interface also shows sections for 'People' (Assignee: Darren Mather), 'Dates' (Created: 3 hours ago), 'Time Tracking' (Estimated: 1d 4h), and 'Agile' (Active sprint: NIW Sprint 1 ends 28/May/18).



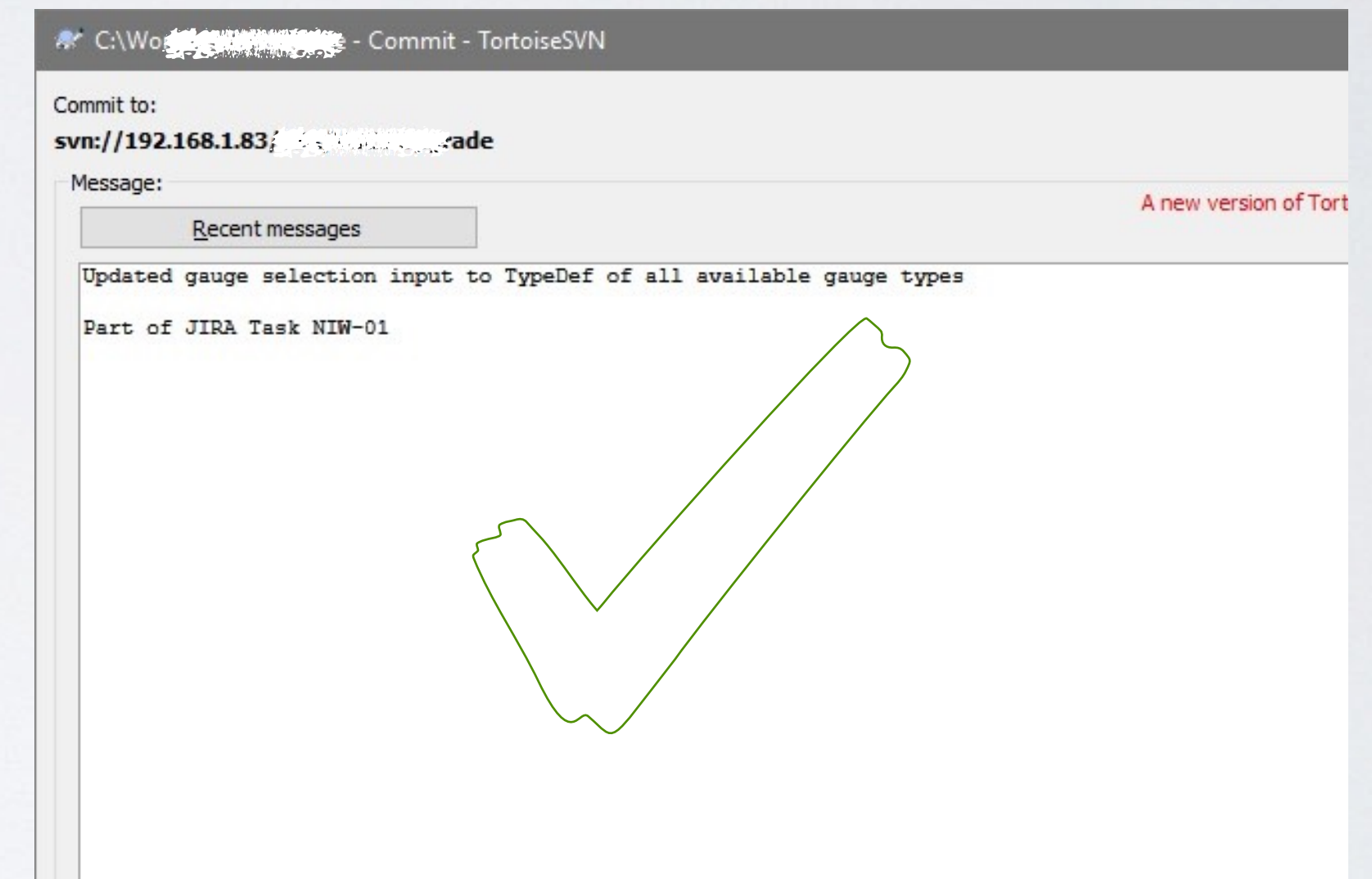
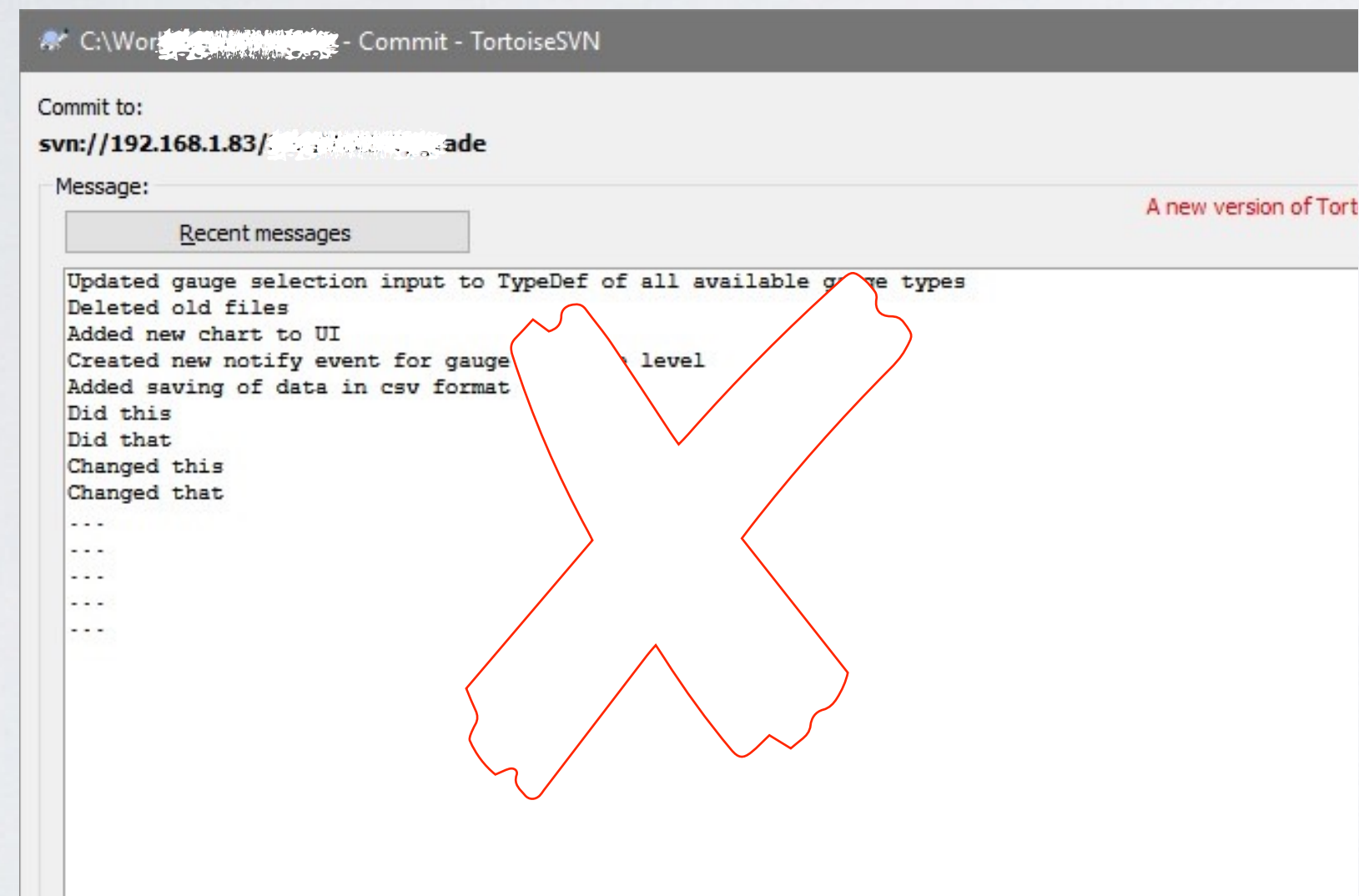
# SOURCE CODE CONTROL - WHY?





# SOURCE CODE CONTROL - USING

Only commit one task at a time





# SOURCE CODE CONTROL - USING

Reference “Task Number” in comments





# SOURCE CODE CONTROL - USING

In JIRA, add commit revision to Task Description/Comment



NIWeek / NIW-1

## Create Alarm Causes LookUp Table

Edit Comment Assign To Do In Progress Done Admin

**Details**

Type: ☒ Task Status: **DONE**  
Priority: **Medium** (View workflow)  
Labels: **Alarms** **UI** Resolution: Done  
Sprint: NIW Sprint 1

**Description**

Use a LookUp table to translate the alarm number/code from the HW into a human readable text string

**Attachments**

Drop files to attach, or [browse](#).

**Activity**

All **Comments** Work log History Activity



There are no comments yet on this issue.

Comment

Style B I U A \*A Link Image List Bulleted List @ +

Fulfilled in SCC Commit rev. 265

**People**

Assignee:  Darren M  
Reporter:  Darren M  
Votes: 0  
Watchers: 1 Stop watching

**Dates**

Created: 3 hours ago  
Updated: Just now  
Resolved: 1 minute ago

**Time Tracking**

Estimated:   
Remaining:   
Logged:

**Agile**

Active sprint: NIW Sprint 1  
[View on Board](#)

**HipChat discussions**

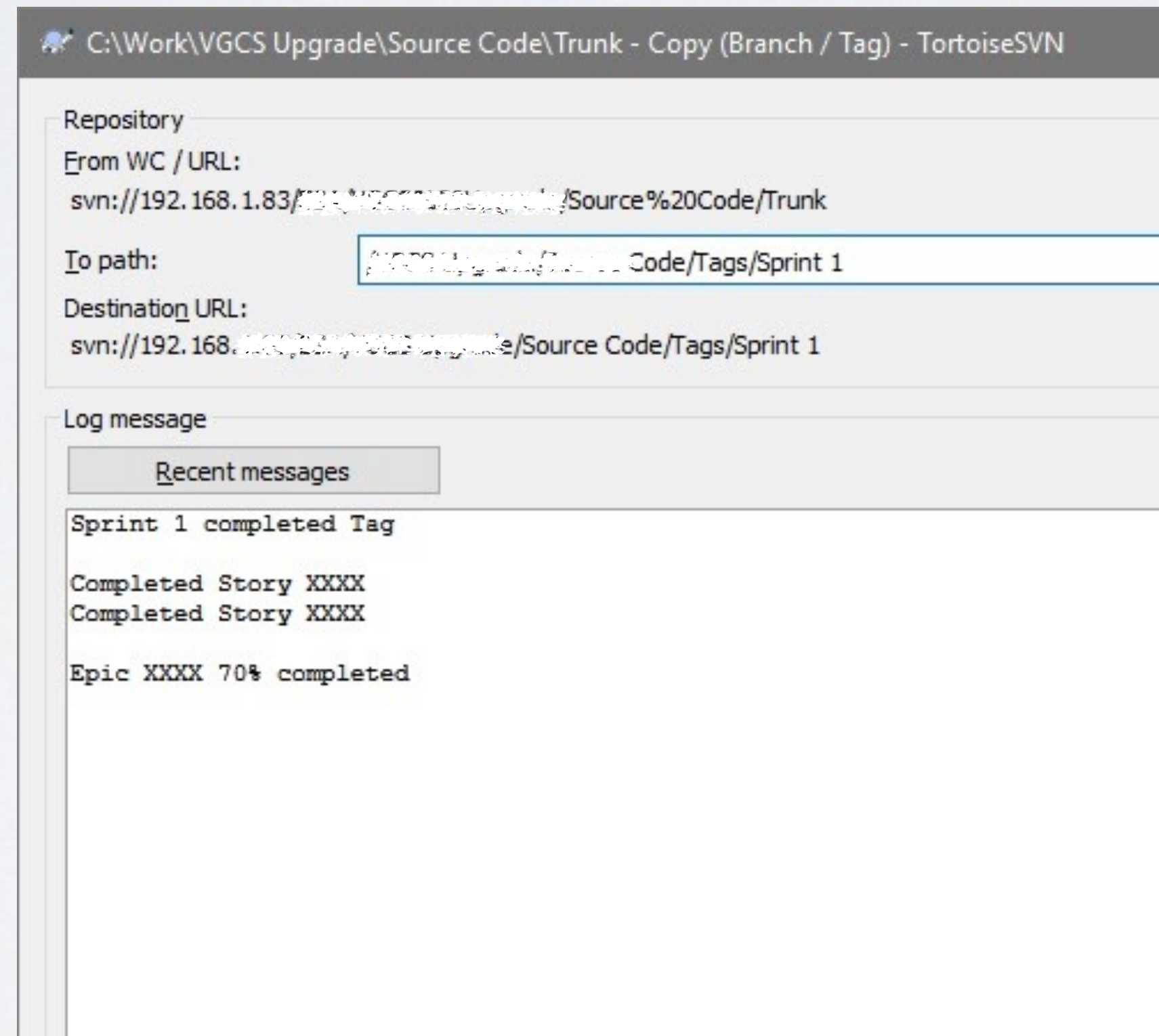
Do you want to discuss this issue? C

[Connect](#) [Dismiss](#)



# SOURCE CODE CONTROL - USING

At the end of each Sprint, create a Tag





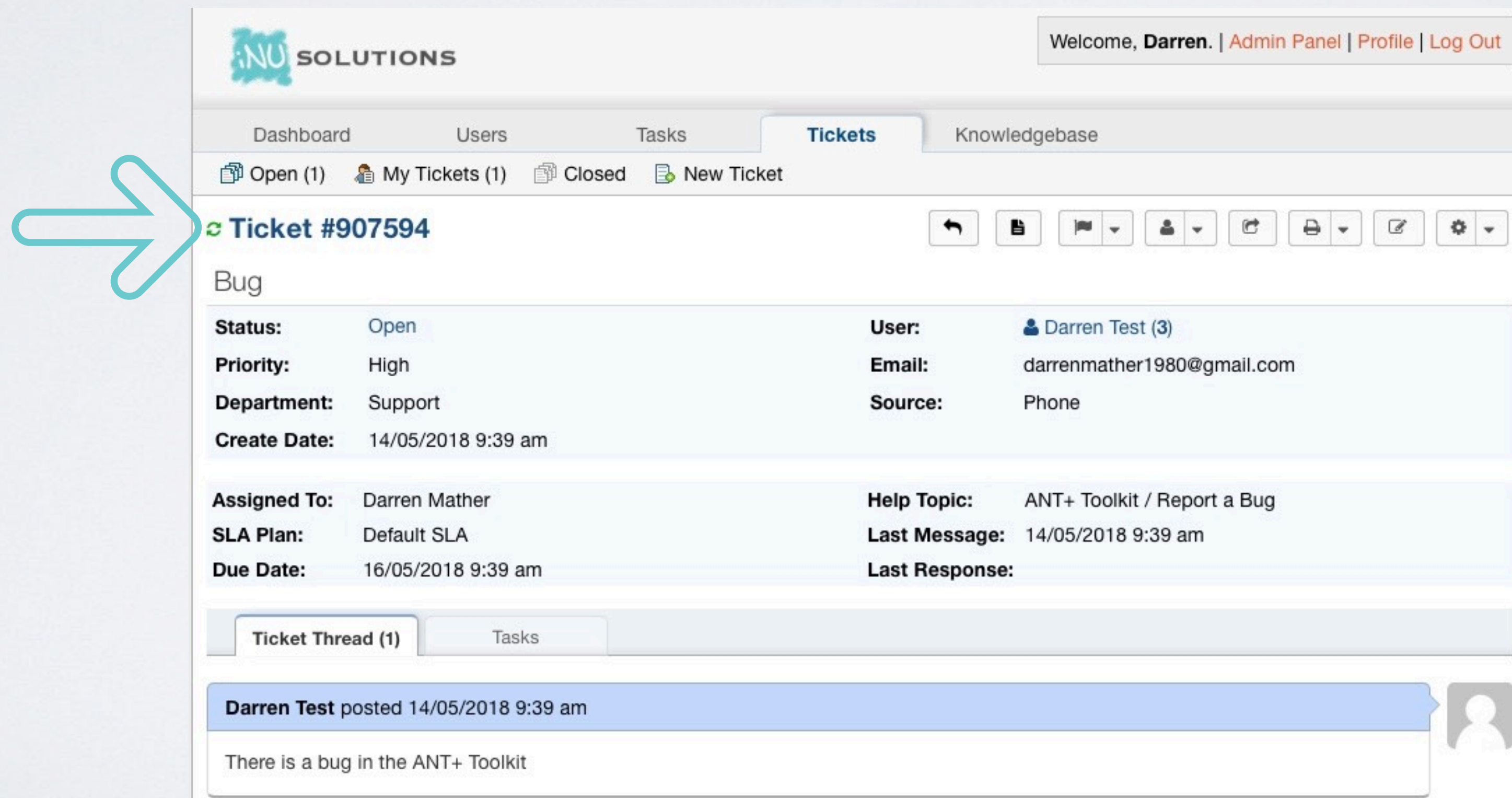
# ISSUE TRACKING - WHY?

- Once released, need some way of managing updates - close the loop
- Issue tracking (or bug tracking) software can be very useful
- Many available platforms, some very good free ones
- Most have a customer side and developer side



# ISSUE TRACKING - USING

Each Bug/Issue is assigned a number



The screenshot shows the iNU SOLUTIONS interface for viewing a specific ticket. A large blue arrow points to the ticket title 'Ticket #907594'. The ticket is categorized as a 'Bug' and is currently 'Open' with a 'High' priority. It was created on 14/05/2018 at 9:39 am by Darren Test (3). The ticket is assigned to Darren Mather, with a default SLA plan and a due date of 16/05/2018 at 9:39 am. The help topic is 'ANT+ Toolkit / Report a Bug'. The last message was sent on 14/05/2018 at 9:39 am. The ticket thread shows a single message from Darren Test: 'There is a bug in the ANT+ Toolkit'.

Dashboard Users Tasks **Tickets** Knowledgebase

Open (1) My Tickets (1) Closed New Ticket

**Ticket #907594**

Bug

**Status:** Open **User:** Darren Test (3)  
**Priority:** High **Email:** darrenmather1980@gmail.com  
**Department:** Support **Source:** Phone  
**Create Date:** 14/05/2018 9:39 am

**Assigned To:** Darren Mather **Help Topic:** ANT+ Toolkit / Report a Bug  
**SLA Plan:** Default SLA **Last Message:** 14/05/2018 9:39 am  
**Due Date:** 16/05/2018 9:39 am **Last Response:**

Ticket Thread (1) Tasks

Darren Test posted 14/05/2018 9:39 am

There is a bug in the ANT+ Toolkit



# ISSUE TRACKING - USING

Add this to JIRA as a “Bug”

osTicket Reference

Type “bug”

osTicket Link

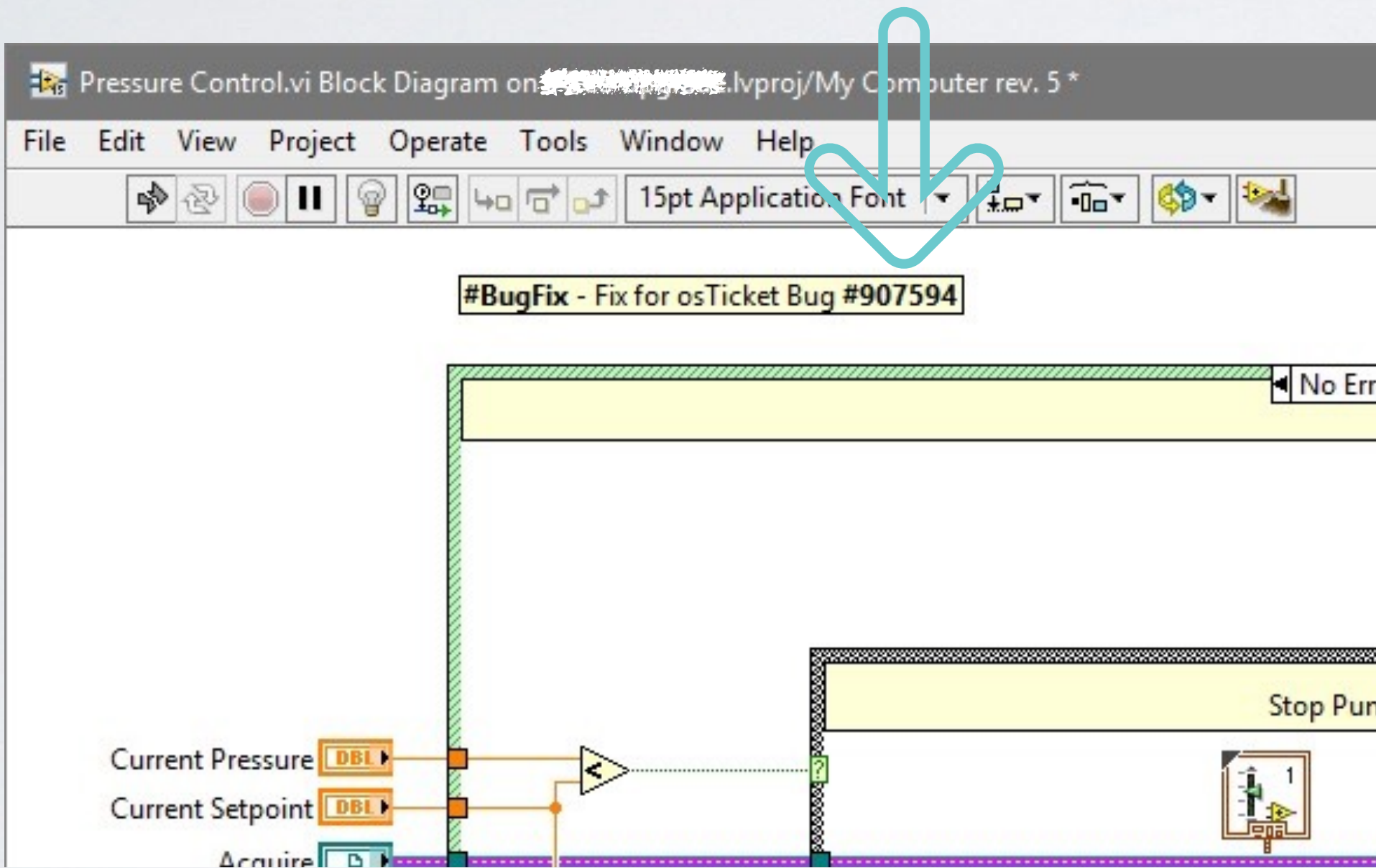
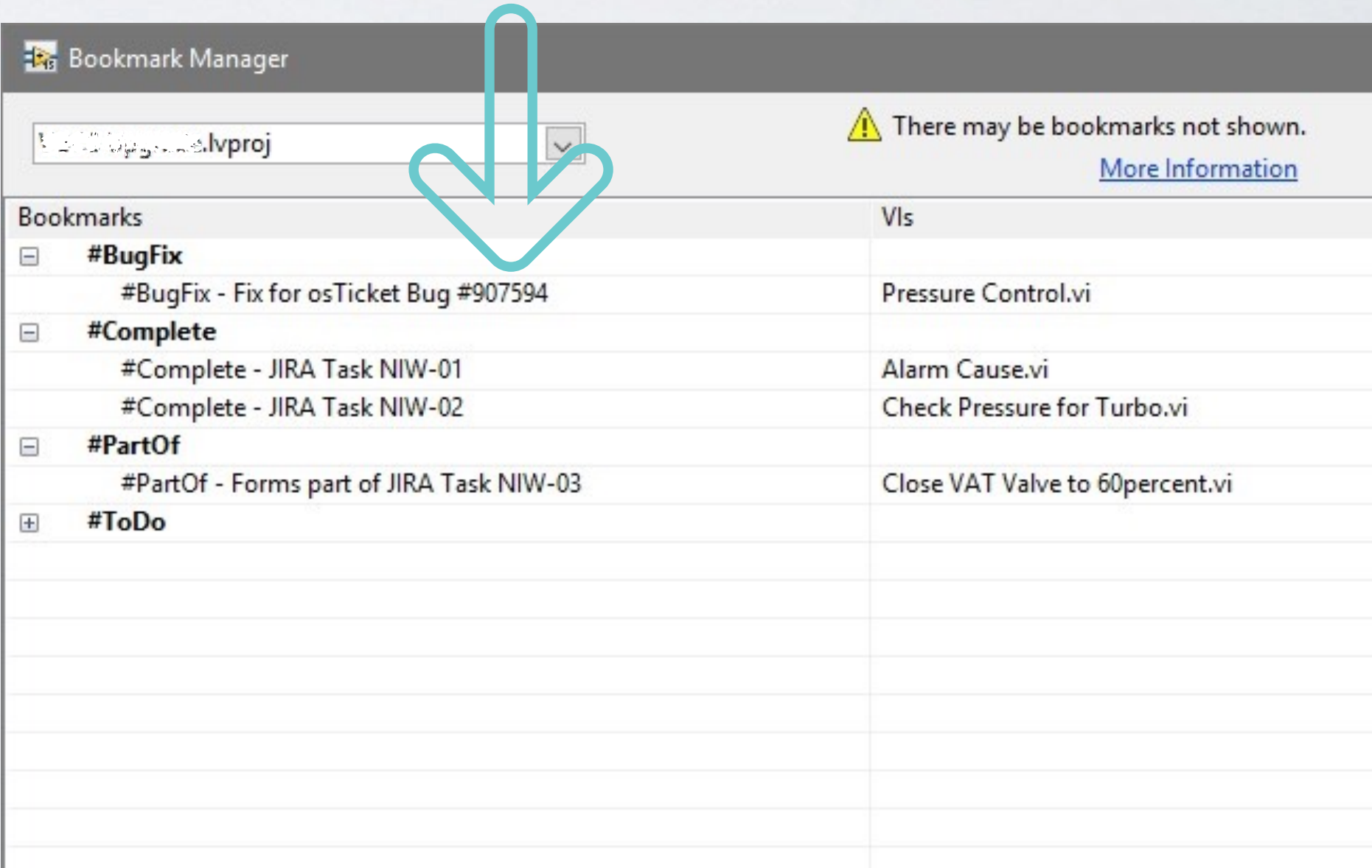
Assigned Dev

The screenshot shows a Jira Software issue page. The issue is titled "Bug from osTicket number 907594". The issue type is "Bug", priority is "Medium", and status is "Unresolved". The description contains a link to the osTicket: "http://inusolutions.com/HelpDesk/scp/tickets.php?id=7". The assignee is Darren Mather. The page also shows fields for "People" (Assignee, Reporter, Votes, Watchers), "Dates" (Created, Updated), "Agile" (View on Board), and "HipChat discussions".



# ISSUE TRACKING - USING

Include the issue number on BD

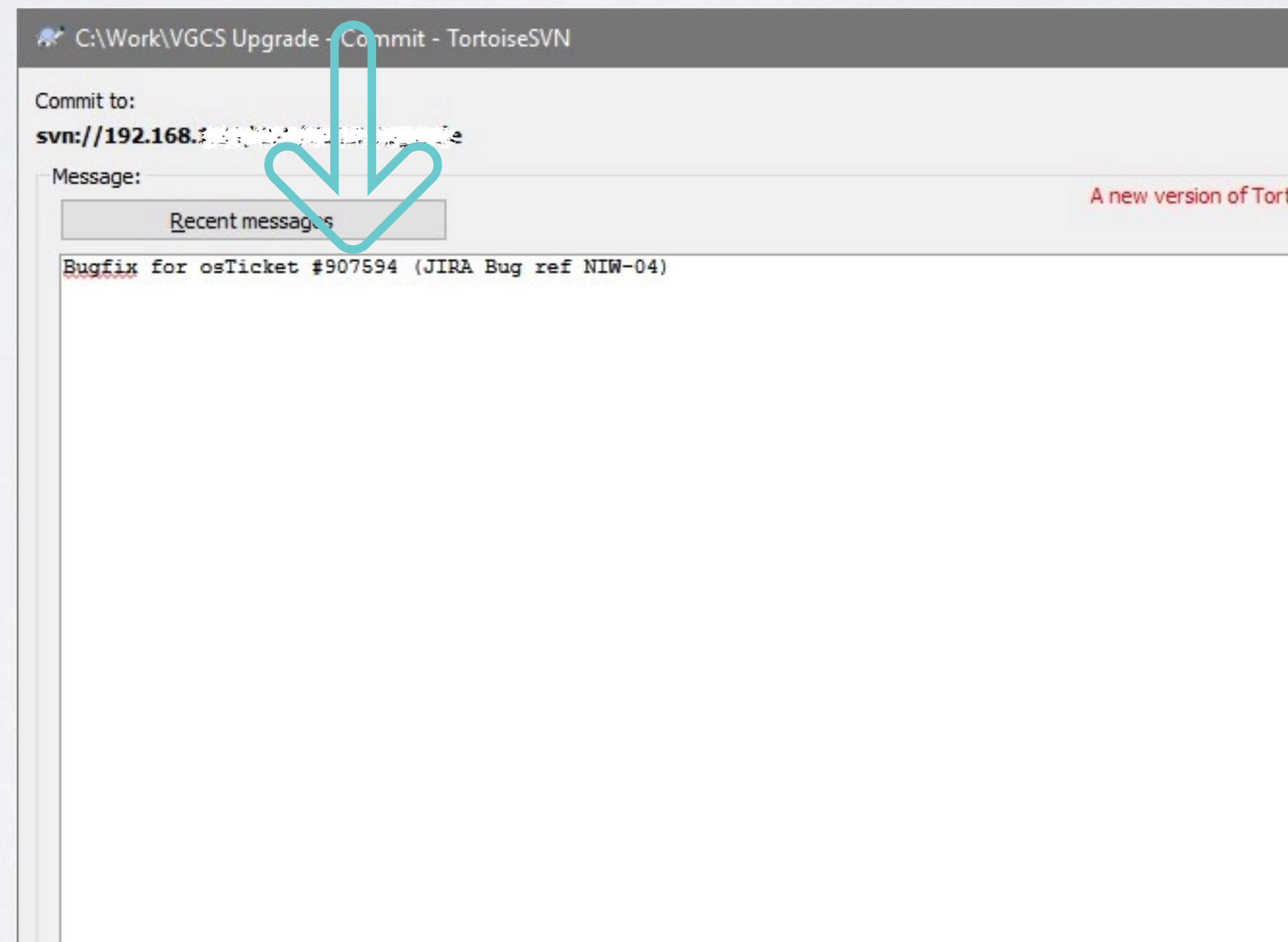
The screenshot shows the 'Bookmark Manager' window. A blue hand icon points to the '#BugFix' bookmark entry. The table below lists the bookmarks and their associated VIs.

Bookmarks	VIs
#BugFix	
#BugFix - Fix for osTicket Bug #907594	Pressure Control.vi
#Complete	
#Complete - JIRA Task NIW-01	Alarm Cause.vi
#Complete - JIRA Task NIW-02	Check Pressure for Turbo.vi
#PartOf	
#PartOf - Forms part of JIRA Task NIW-03	Close VAT Valve to 60percent.vi
#ToDo	



# ISSUE TRACKING - USING

Include issue number in commit description





# TESTING

- Many different schools of thought on how to test
  - Unit Testing
  - Continuous Integration Testing
- Different test processes for different scenarios
  - Factory Acceptance Test (FAT)
  - Site Acceptance Test (SAT)
  - Compatibility Testing



# TESTING

- Important that whichever method you use/choose, it integrates into your work process
- If you are using JIRA and Bitbucket then it makes sense to also use Bamboo
- Most open source tools have a command line instruction set
  - Can use this to do advanced/automated tasks not in the UI
- Some industries require certain testing strategies (medical, military etc.)
- If you use Unit Testing then re-validating code becomes easier

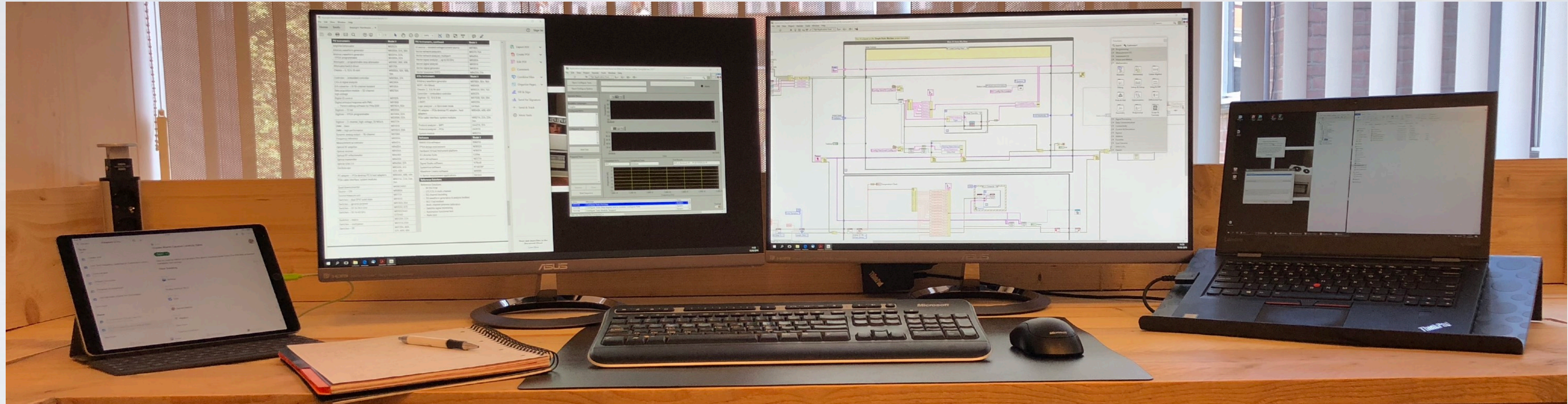


# TESTING

- Developing a rugged “capture all” testing process is not small task
- Can be iterating development for years and still not get 100% coverage
- Use Pareto (and a bit of common sense) to determine what can be “missed”
- There are many tools already available, check the LVTN



# DESK LAYOUT



- iPad
  - JIRA
  - Slack
  - osTicket
- Left Screen
  - Front Panels
  - Documentation
  - Anchored
- Right Screen
  - Block Diagrams
  - Maximised
  - Pallets
- Laptop Screen
  - Project Explorer
  - Error List
  - Context Help
  - File Explorer



# DESK LAYOUT





# OFFICE LAYOUT

- Organise your office to make working easier
- Create different “work areas” for different tasks (admin, tooling, etc.)
- Try and have project HW setup and connected all the time
- Use whiteboards, chalk boards, touch screens etc. to visualise things
- Never underestimate the power of a comfy chair and cuppa!



# OFFICE LAYOUT





# TOOLS

- Built-in tools
  - Bookmark Manager (free)
  - VI Analyser (Free in Pro)
  - Requirements Gateway (Paid)
- Add-ons
  - TSVN from Viewpoint Systems
  - Caraya & VI Tester from JKI
  - On LVTN there is a “Software Engineering Tools” category



# REQUIREMENTS TRACKING - ADVANCED

- JIVE (JIRA and LabVIEW...!)
  - Can link JIRA to LabVIEW
  - Available soon on the LVTN (Free)
- Open Services for Lifestyle Collaboration (OSLC)
  - Can create a link between LabVIEW and various SaaS platforms
  - Research project by ULMA Embedded Solutions



# ISSUE TRACKING - ADVANCED

- Possible to create a link between Zendesk and OpenProject
  - Uses APIANT cloud based platform
  - Not free
- osTicket can be linked with
  - Trello
  - Slack
  - Can use S3 to mark bugs as “fixed” direct from LabVIEW (maybe...)



# ADDITIONAL RESOURCES

- Presentations
  - Practical Techniques: Gathering and Managing System Requirements
    - Becky Linton & Jarobit Pina (SEPAD Track at NI Week 2018)
  - Organise Your Code Well or You'll Screw Up Your Project
    - Chris Cilino & Deborah Burke (SEPAD Track at NI Week 2018)
- Atlassian Agile Coach
  - <https://www.atlassian.com/agile>



# SOFTWARE OPTIONS

- SCC
  - Subversion
  - BitBucket
  - Git
- Requirements Tracking
  - JIRA
  - Doors
  - OpenProject
  - Jama
  - PlanIO
- Bug Tracking
  - osTicket
  - ZenDesk
  - JIRA Service Desk
  - Bugzilla
- Development Methodologies
  - Agile
  - Spiral
  - Waterfall



# SUMMARY

- A well defined and “useable” process should not be seen as an overhead
- Can be used to reduce risk, increase productivity and accurately track progress
- Many tools out there to help, some integrate better than others, but it’s getting better
- The community is starting to drive innovation from SaaS providers



*See you at GDevCon!*



Cambridge UK  
4-5<sup>th</sup> September 2018

Check your inbox / Twitter / LinkedIn for ticket information later this week