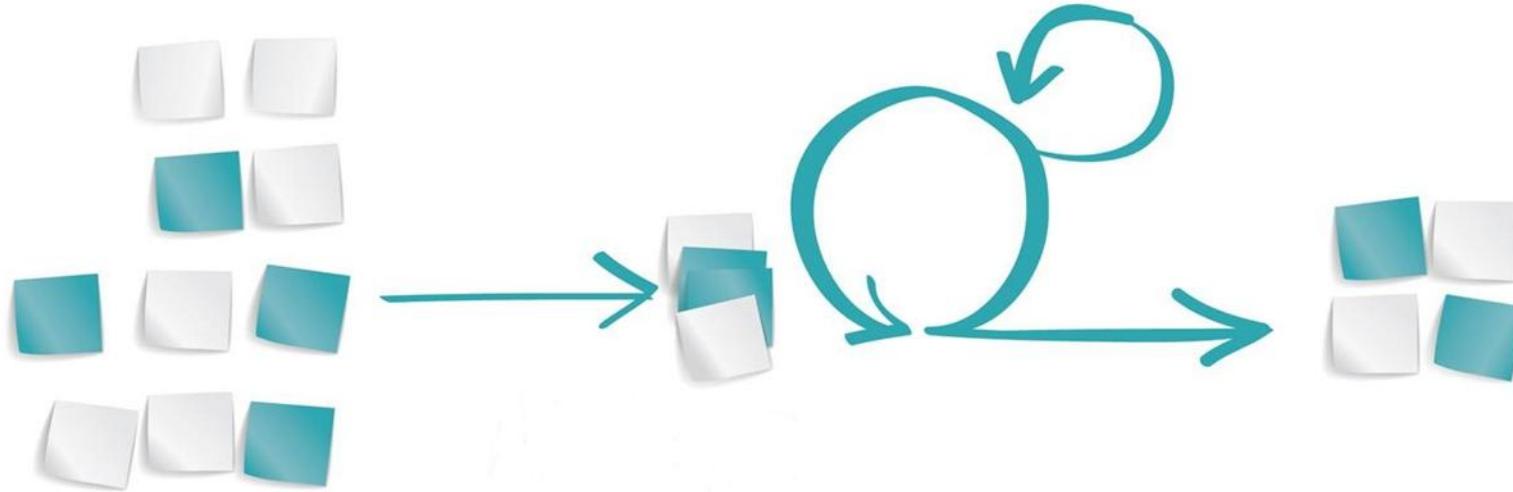




**intelliware**  
software development



# Agile Story Writing



## What You'll Learn in this Presentation:

- The basics of user stories.
- How user stories fit into the overall Agile planning process.
- How to write a user story.

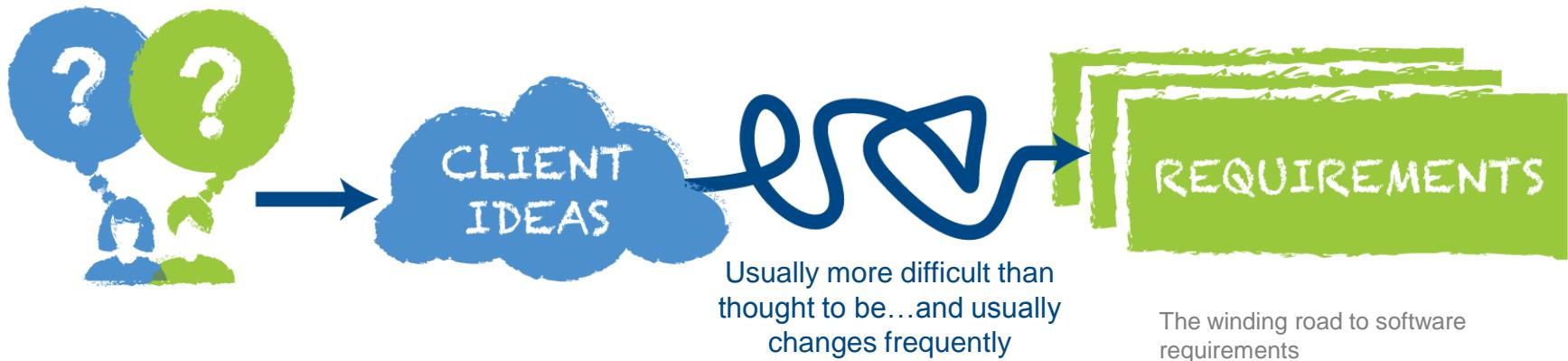
### Show Account Balance

As a Private Investor, show me my current account balance so I can make decisions on future transactions that will affect my account.

A story card example

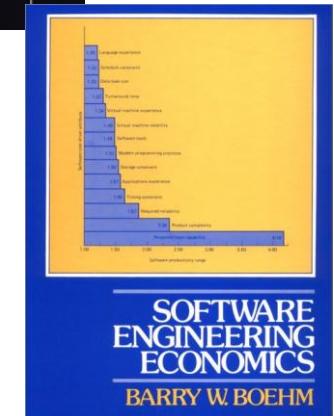
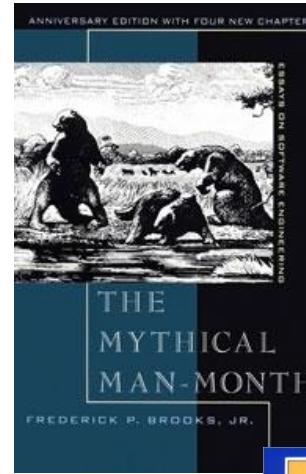
## Why is it so Difficult to Determine Software Requirements?

- Requirements gathering is when informal ideas become formal concepts:
  - Converting a concept into something concrete is almost always more difficult than it is initially believed to be.
  - The concept of what the concrete version needs to look like changes frequently.



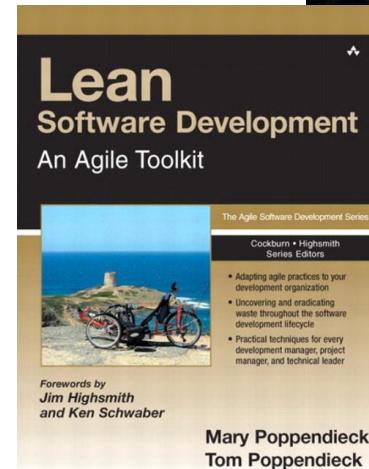
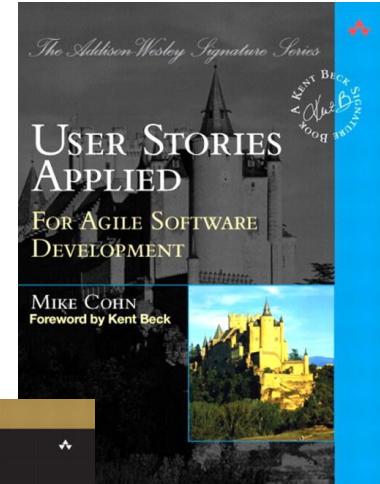
## Why are Requirements So Important?

- According to Fred Brooks (the author of *The Mythical Man Month*):
  - “*The hardest single part of building a software system is deciding precisely what to build. No other part is as difficult...No other part of the work so cripples the resulting system if done wrong.*”<sup>1</sup>
- According to Barry Boehm (*Software Engineering Economics*) and other software engineering experts, around 75-80% of all errors found in software projects can be traced back to the design and requirements phases.<sup>2</sup>



# The Challenges with Written Requirements

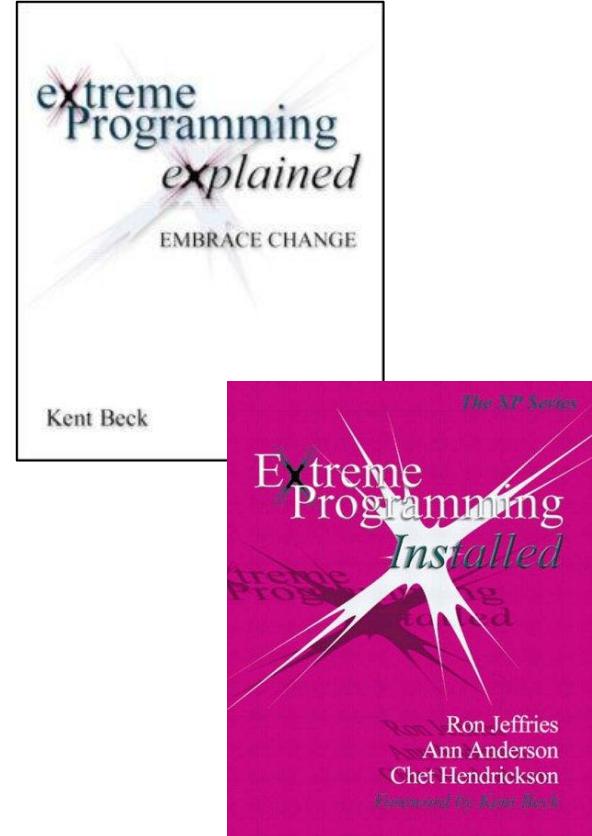
- According to Mike Cohn, author of *User Stories Applied*:
  - “*Writing things down is no guarantee that customers will get what they want; at best they’ll get what they wrote down.*”<sup>3</sup>
- From *Lean Software Development* by Mary & Tom Poppendieck, the Seven Wastes of Software Development:<sup>4</sup>
  1. Partially Done Work
  2. **Extra Processes (paperwork)**
  3. Extra Features
  4. Task Switching
  5. Waiting
  6. Motion
  7. Defects



## What is a User Story?

There are numerous definitions for stories.

- A common definition:
  - ***A short description of a function that an end-user would want.***
- From Kent Beck:
  - ***“One thing the customer wants the system to do... (it) should be testable.”***<sup>5</sup>
- From Ron Jeffries:
  - ***“Stories are promises for... the series of conversations that will take place between the customer and the programmers.”***<sup>6</sup>



## Why User Stories?

Stories have many advantages.

- **Easy to understand**

- Written in non-technical language that customers / product owners can relate to.

- **Work at the right level**

- Not too detailed, are easy to manipulate and move around, like a deck of cards.

- **Relatively easy to create**

- Writing stories takes some skill, but experts can define entire systems for planning purposes in a matter of hours.



# Story Types

## Epic

- Represents multiple features or many stories.
- Can take months to build and works at the release level.
- What the end users tend to focus on.

## Feature

- Smaller than epics, but bigger than stories.
- Can take weeks, possibly one or more iterations to build.
- What customers / product owners tend to focus on.

## User Story

- Are the smallest increment of value.
- Take days, perhaps a week or two at most to build.
- What development teams tend to focus on.



Primary focus of this presentation.

# Key Players: Customers & Developers

## 1. Customers / product owners

- The people who know how to do what the system is going to be doing.
- They either are the end user or they are representative of the eventual user of the system.

## 2. Programmers

- The people who will be building, testing, deploying, documenting & training those who will use the system.

Stories are key to fulfilling the requirements in the **Customer and Programmers Bills of Rights**.

### Customer Bill of Rights

- You have the right to an overall plan
- You have the right to change your mind, to substitute functionality, and to change priorities
- You have the right to be informed of schedule changes, in time to choose how to reduce scope to restore the original date

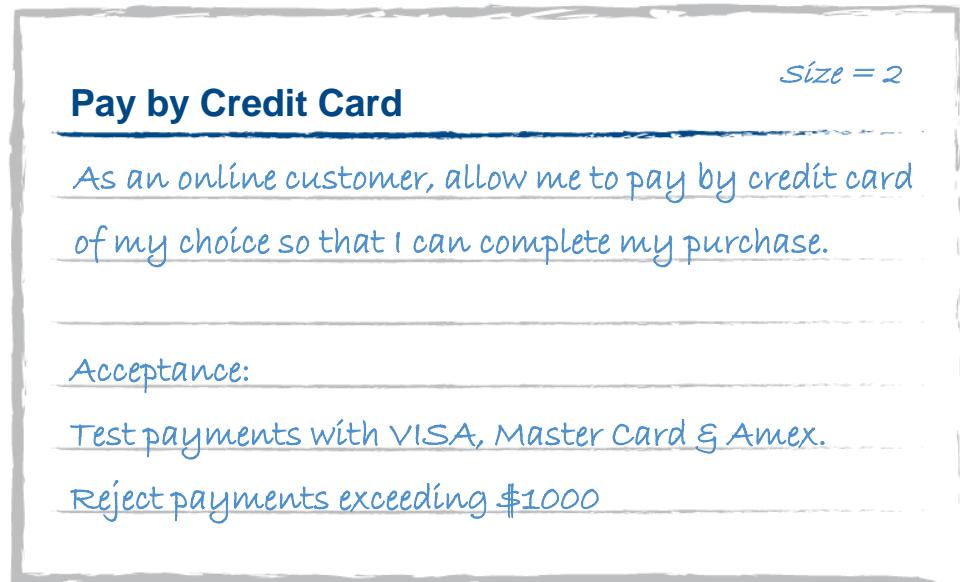
### Programmer Bill of Rights

- You have the right to know what is needed, with clear declarations of priority
- You have the right to make and update your own estimates

## User Story Components

At minimum, a user story has:

- 1. A card** to write the story on.
- 2. A name** that the customers and developers understand.
- 3. A description** (should be limited to one or two sentences).
- 4. Acceptance criteria** to define when the story will be considered completed.
- 5. A size estimate** for time management.



A typical story card.

# User Story Components

Client Logo (optional)	intelliware.ca software development	Story Priority (optional)	Story Size
		Priority <b>2</b>	Size <b>1</b>
Story Name	<b>Login</b>		
Description (no more than 2-3 sentences)	<p><b>Description:</b> As an authorized user, provide a login page to gain access to the secure pages of the Web site.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"><li>• Users to provide IDs and Passwords.</li><li>• A 'Forgot your password?' link will be needed.</li></ul>		
Story Notes (optional)			
Acceptance (list of criteria to indicate when the story will be closed)	<p><b>Acceptance:</b> To be provided by the Customer. .</p>		
		ID <b>SMP-1-2</b>	
		Release, Module and ID Fields (optional - help to identify the story)	

## Story Actions

Stories can be:

- **Split**

- **Split**
  - A large story can be split into two or more smaller ones of different sizes; useful for breaking up epics.

- **Combined**

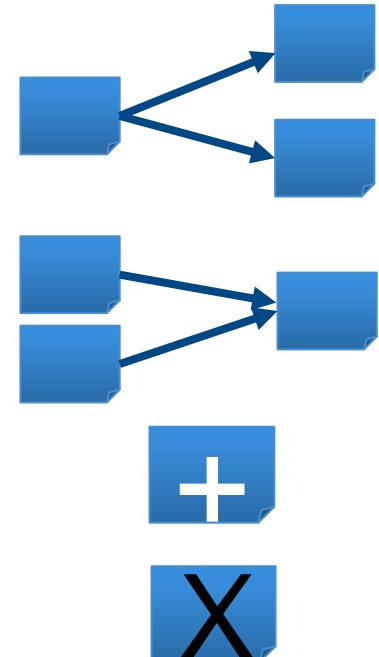
- **Combined**
  - Two or more small stories can be combined into one.

- **Added**

- **Added**
  - New stories can be added to an existing backlog.

- **Deleted**

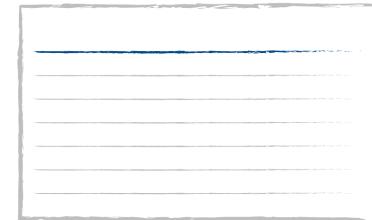
- **Deleted**
  - Existing stories can be deleted from a backlog.



## The 3 Cs of Stories

- **C**ard

- A token to represent some customer functionality.
- Stories represent customer requirements rather than document them.
- Using a card keeps the story short.



- **C**onversation

- Customers and developers discuss the details of the story at the time it is to be developed, not before then.



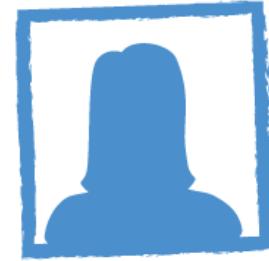
- **C**onfirmation

- The customer should provide acceptance tests for the story, and then see them run to confirm that the story has been completed.



## User Roles and Description Formats

- **Identifying user roles helps with writing stories.**
- **Standard story description template:**  
*As a [role], provide [function] so that [business value].*
- **Some simple examples:**
  - As a customer, provide a button that I can use so that I can connect directly with the call centre when my order gets stuck.
  - As a call centre rep., review orders in progress online so that I can help customers complete their orders.
  - As a manager, access stats on incomplete online orders so that I can make decisions on how to improve the ordering process.



Call Centre Rep.



Manager



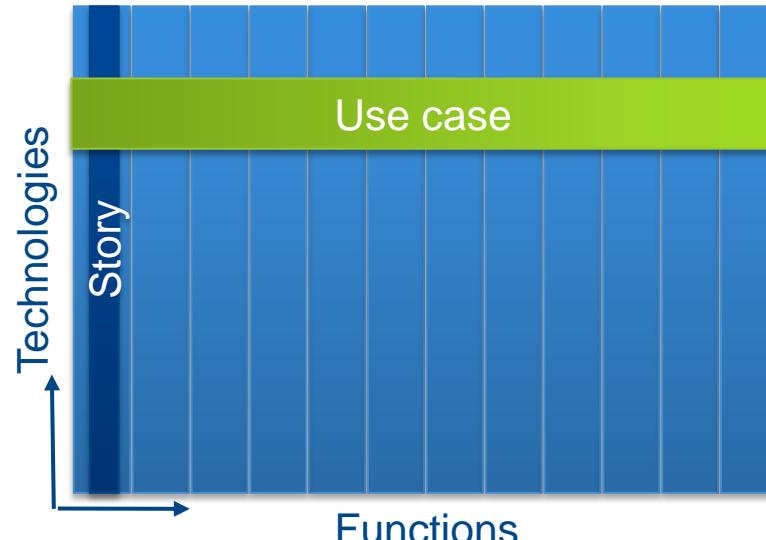
Customer

## Writing Stories – The INVEST Framework<sup>7</sup>

- **I**ndependent – Dependencies between stories should be avoided.
- **N**egotiable – Stories should be written so that the details can be negotiated in a conversation between the customer and the development team.
- **V**aluable – The feature should have business value to the customer.
- **E**stimatable – Stories should be understood well enough by customers and should be small enough to be “estimatable”.
- **S**mall – Stories that are too big are not useful in planning.
- **T**estable – It must be possible for the development team to write tests for the story.

## User Stories vs. Use Cases

- A **story** can be considered similar to a lightweight use case.
  - A story can represent a small piece of a use case.
- **Use cases** cut across many functions and may touch on many stories.



## Examples of Common Story Mistakes

### Easy to Use

As any user of the system, make sure the user interface is easy to use so that I can be more efficient, make more purchases and will be more likely to return.

Not testable!

Not functional.

## Examples of Common Story Mistakes

### Pay By Credit Card

*As a customer, pay by VISA, MasterCard and American Express.*

Too big...should probably be split as follows:

1. Pay by one credit card (including payment infrastructure)
2. Pay by additional credit cards

## Examples of Common Story Mistakes

### System Administration

*As a power user or manager, provide a user interface to facilitate management of the system.*

This is huge, with many functions; an epic!

Also, there's more than one user type.

## Examples of Common Story Mistakes

### Choose Flight

*As a customer, select a flight from the list presented by the List Flights Story and confirm my selection.*

This is not independent as there is a dependency on the List Flights Story.

Should be split along another dimension.

## For More Information

Mike Cohn's site contains a good section on user stories:

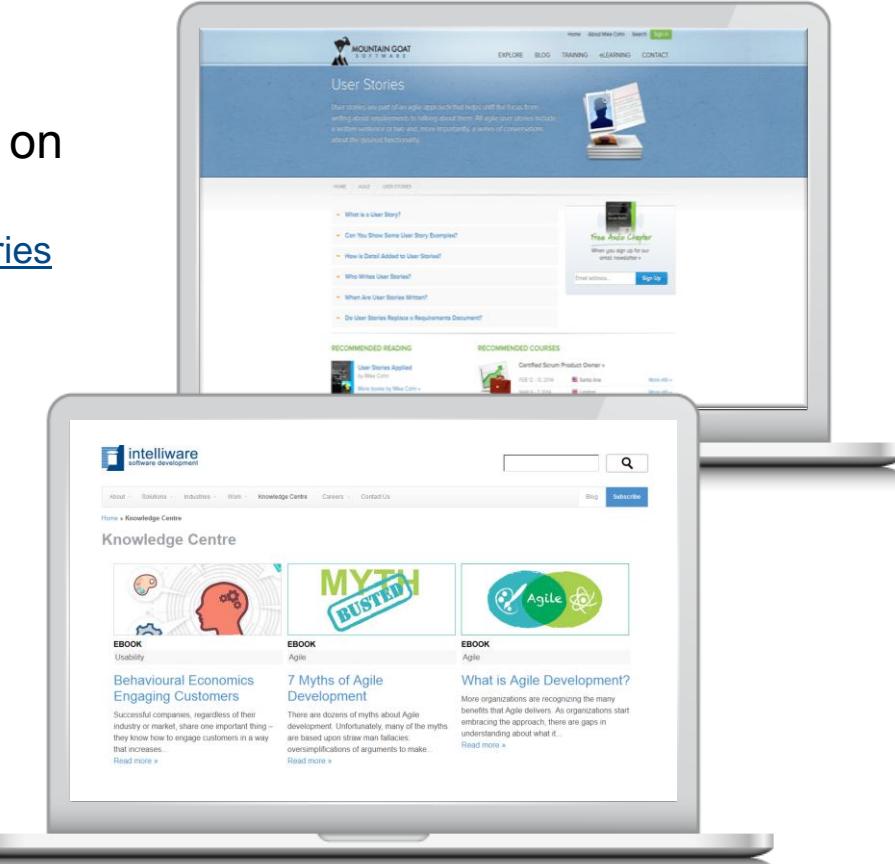
<http://www.mountaingoatsoftware.com/agile/user-stories>

The Agile Alliance site is also a good resource:

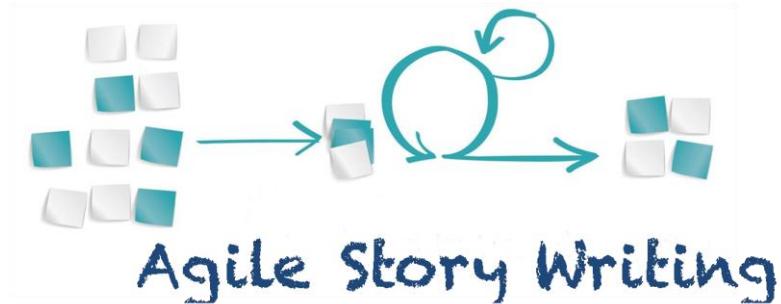
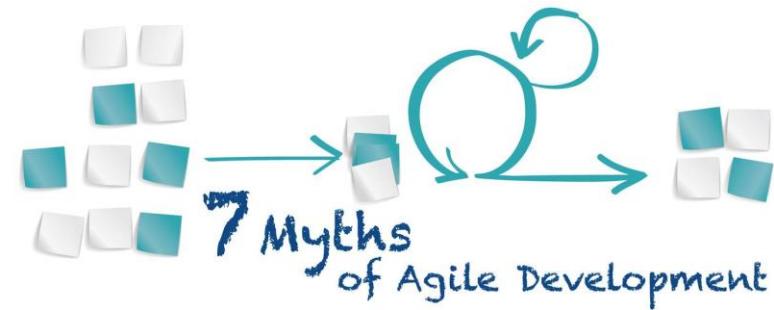
<http://guide.agilealliance.org/guide/user-stories.html>

Intelliware's Knowledge Centre contains several resources on the basics of Agile:

<http://www.intelliware.com/knowledge-centre>



## Check Out Other Titles From Our Agile Development Series



## About Intelliware Development Inc.

Intelliware is a custom software, mobile solutions and product development company headquartered in Toronto, Canada. Intelliware is a leader in Agile software development practices which ensure the delivery of timely high quality solutions for clients. Intelliware is engaged as a technical partner by a wide range of national and global organizations in sectors that span Financial Services, Healthcare, ICT, Retail, Manufacturing and Government.



[/company/intelliware-development-inc-](#)



[/intelliware.inc](#)



[/intelliware\\_inc](#)



[/GooglePlusIntelliware](#)

[www.intelliware.com](#)