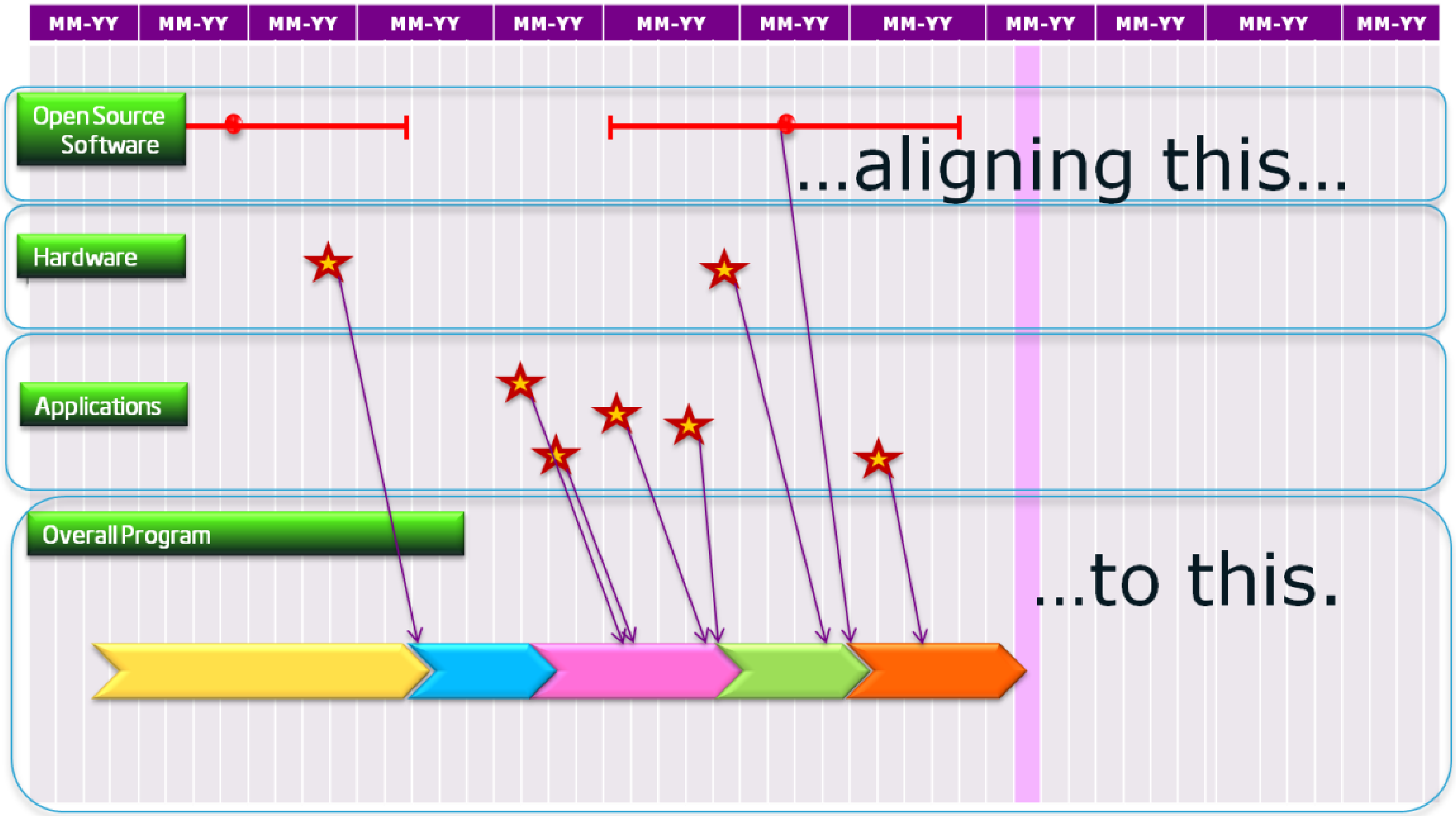
Delivering Quality and Predictability *While Embracing the Open Source Spirit*

August 17th, 2011

Julie Fleischer, Intel Corporation

INTEL OPEN SOURCE TECHNOLOGY CENTER

We Will Cover…



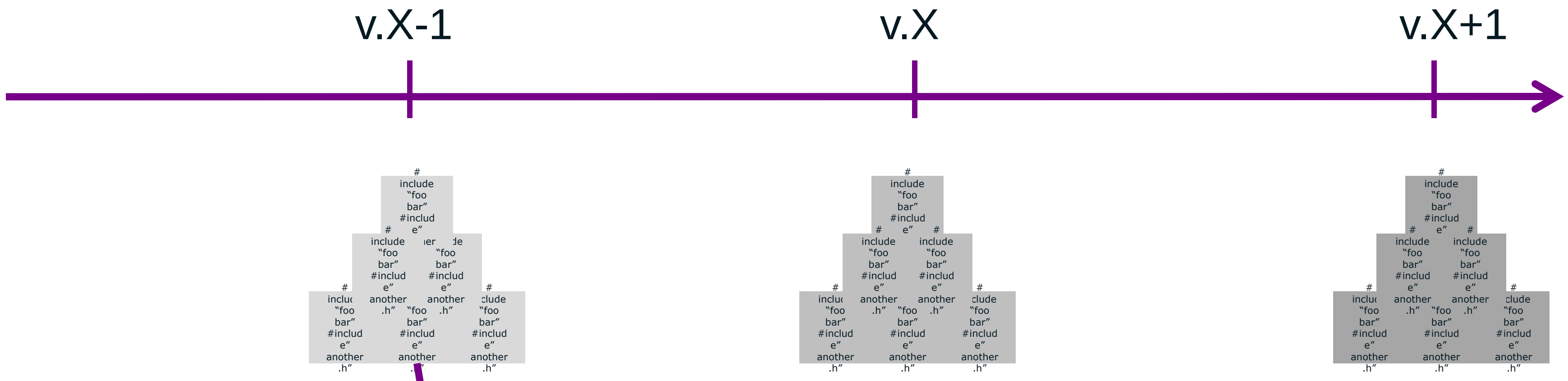
The Tips on Delivering Quality

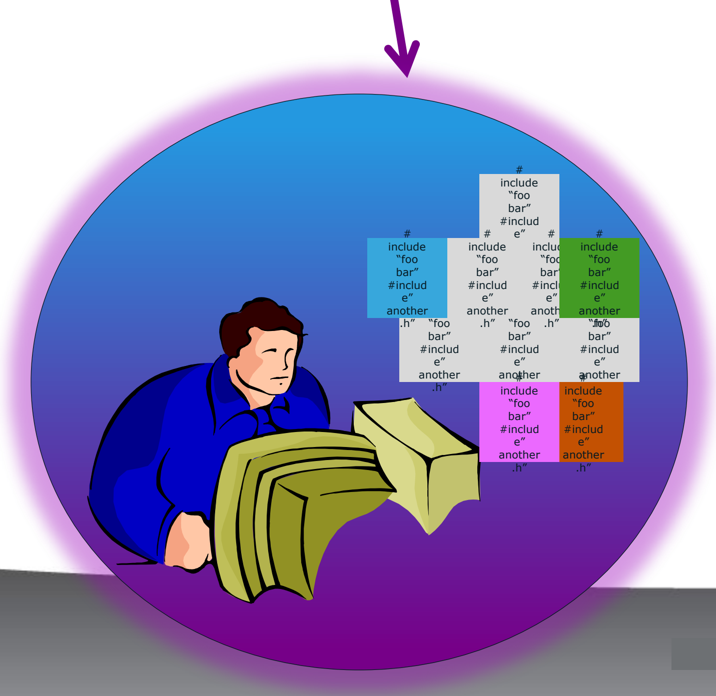
- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

The Tips on Delivering Quality

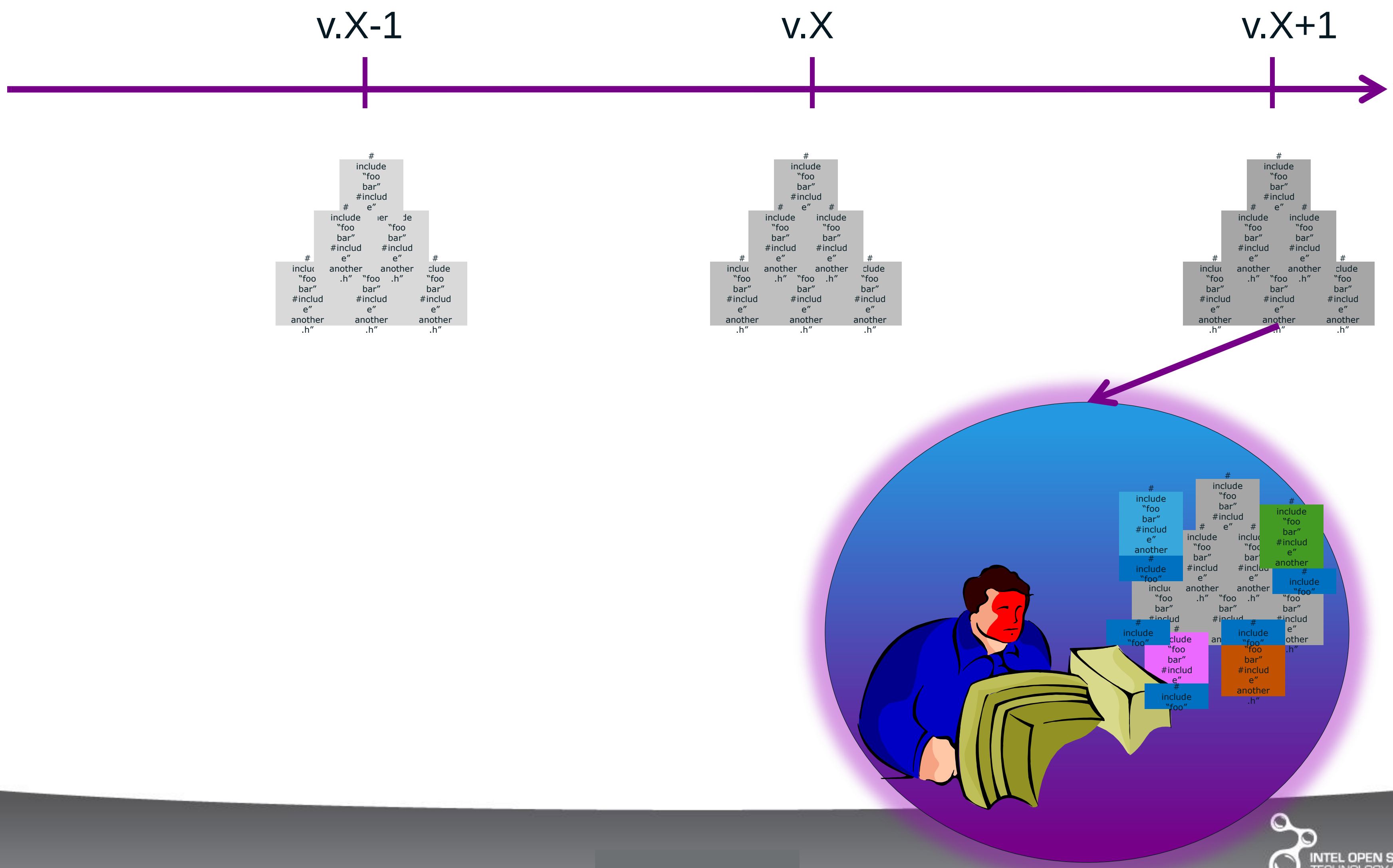
- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

An Open Source Project has its own timeline…

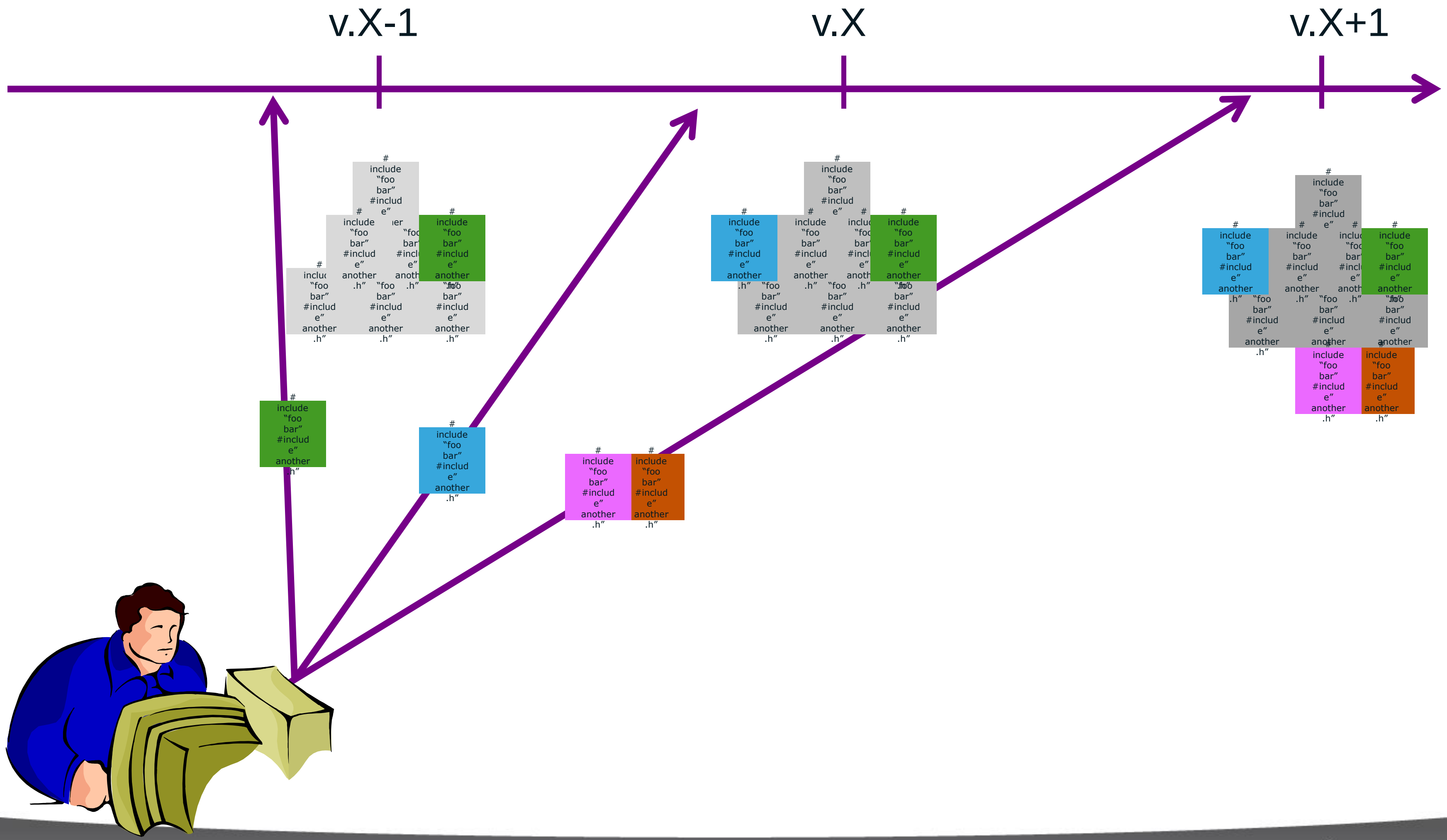




If you develop outside of the timeline…



Tip #1: Leverage the open source environment



The Tips on Delivering Quality

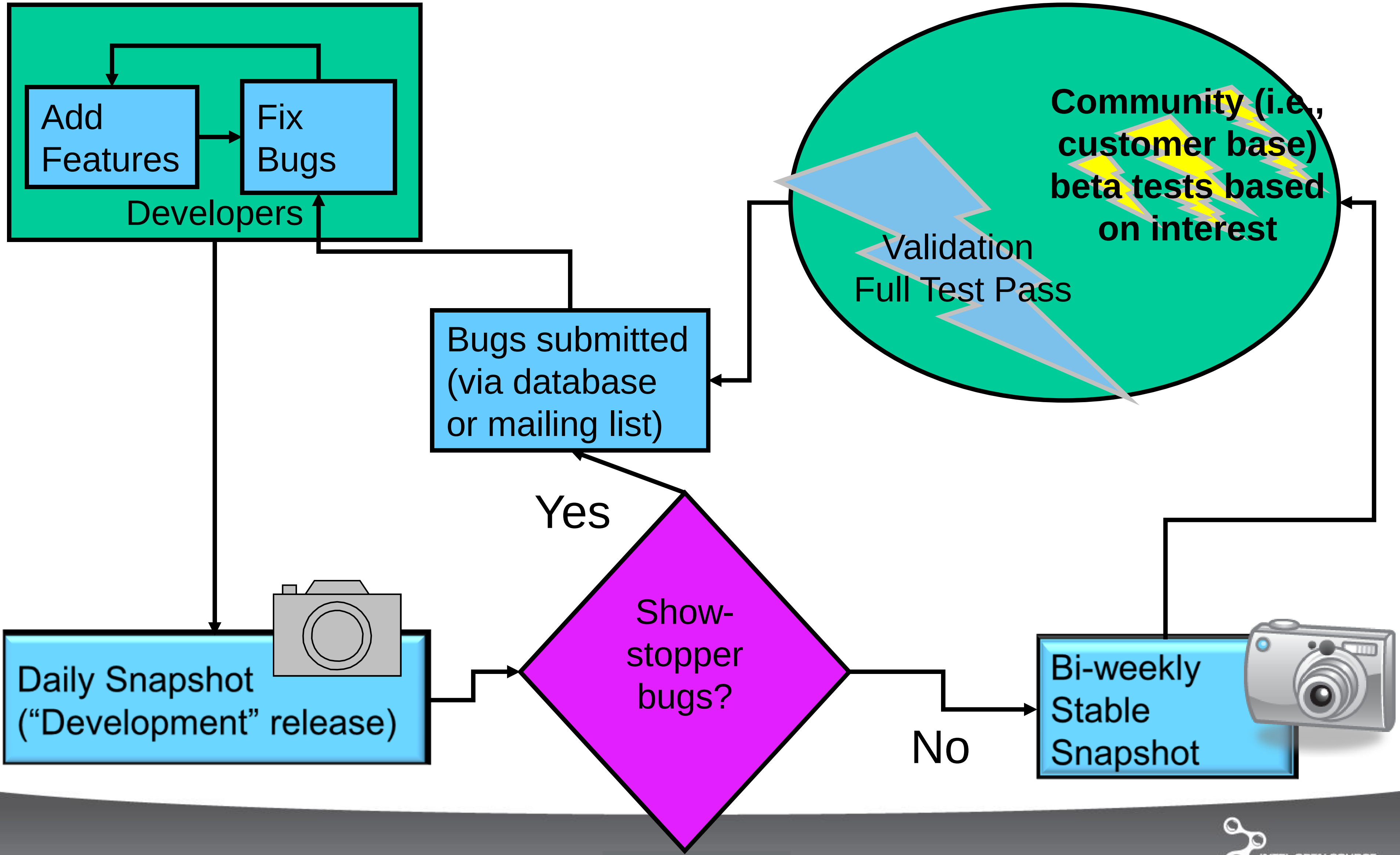
- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

When I was Validation Lead for an exciting open source project…

We had an active open source community and could easily embrace the mantras:

- Release early, release often.
- Give enough eyes, all bugs are shallow.

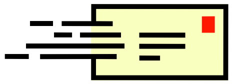
Tip #2: Leverage the open source community


Add Features
Fix Bugs
Developers
Community (i.e., customer base) beta tests based on interest
Validation
Full Test Pass
Bugs submitted (via database or mailing list)
Yes
Show-stopper bugs?
No
Daily Snapshot ("Development" release)
Bi-weekly Stable Snapshot

The Tips on Delivering Quality

- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

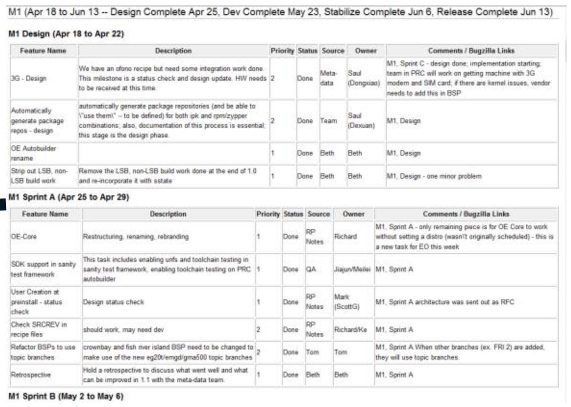
This open source project had a solid requirements definition process…



Define requirements

Prioritize requirements

Put requirements in Wiki



M1 (Apr 18 to Jun 13 -- Design Complete Apr 25, Dev Complete May 23, Stabilize Complete Jun 6, Release Complete Jun 13)

M1 Design (Apr 18 to Apr 22)

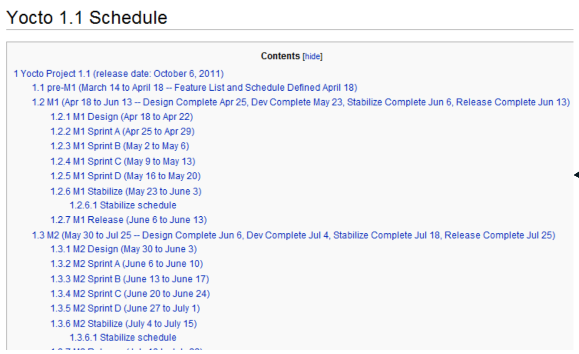
Feature Name	Description	Priority	Status	Source	Owner	Comments / Bugzilla Links
3G - Design	We have an ofono recipe but need some integration work done. This milestone is a status check and design update. HW needs to be received at this time.	2	Done	Meta-data	Saul (Dongxiao)	M1, Sprint C - design done; implementation starting; team in PRC will work on getting machine with 3G modem and SIM card; if there are kernel issues, vendor needs to add this in BSP
Automatically generate package repos - design	automatically generate package repositories (and be able to \"use them\" -- to be defined) for both ipk and rpm/zypper combinations; also, documentation of this process is essential; this stage is the design phase.	2	Done	Team	Saul (Dexuan)	M1, Design
OE Autobuilder rename		1	Done	Beth	Beth	M1, Design
Strip out LSB, non-LSB build work	Remove the LSB, non-LSB build work done at the end of 1.0 and re-incorporate it with sstate	1	Done	Beth	Beth	M1, Design - one minor problem

M1 Sprint A (Apr 25 to Apr 29)

Feature Name	Description	Priority	Status	Source	Owner	Comments / Bugzilla Links
OE-Core	Restructuring, renaming, rebranding	1	Done	RP Notes	Richard	M1, Sprint A - only remaining piece is for OE Core to work without setting a distro (wasn\'t originally scheduled) - this is a new task for EO this week
SDK support in sanity test framework	This task includes enabling unfs and toolchain testing in sanity test framework, enabling toolchain testing on PRC autobuilder	1	Done	QA	Jiajun/Meilei	M1, Sprint A
User Creation at preinstall - status check	Design status check	1	Done	RP Notes	Mark (ScottG)	M1, Sprint A architecture was sent out as RFC
Check SRCREV in recipe files	should work, may need dev	2	Done	RP Notes	Richard/Ke	M1, Sprint A
Refactor BSPs to use topic branches	crownbay and fish river island BSP need to be changed to make use of the new eg20t/emgd/gma500 topic branches	2	Done	Tom	Tom	M1, Sprint A When other branches (ex. FRI 2) are added, they will use topic branches.
Retrospective	Hold a retrospective to discuss what went well and what can be improved in 1.1 with the meta-data team.	1	Done	Beth	Beth	M1, Sprint A

M1 Sprint B (May 2 to May 6)

Create and track schedule



Yocto 1.1 Schedule

Contents [hide]

I didn’t try to change this, I just tried to <u>understand</u> it.

Then, we could add…

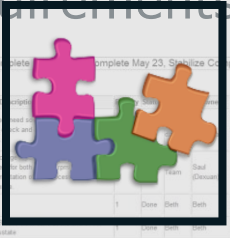
Define requirements

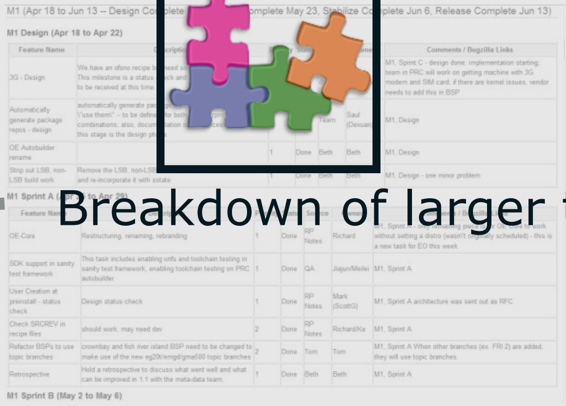
Community input

Prioritize requirements

Developer input

Put requirements in Wiki





M1 (Apr 18 to Jun 13 -- Design Co[illegible]omplete May 23, Stabilize Co[illegible]plete Jun 6, Release Complete Jun 13)

M1 Design (Apr 18 to Apr 22)

Feature Name	Description	Priority	Status	Source	Owner	Comments / Bugzilla Links
3G - Design	We have an ofono recipe b[illegible]need so[illegible] This milestone is a status [illegible]ck and [illegible] to be received at this time	[illegible]	[illegible]	[illegible]	[illegible]	M1, Sprint C - design done; implementation starting; team in PRC will work on getting machine with 3G modem and SIM card; if there are kernel issues, vendor needs to add this in BSP
Automatically generate package repos - design	automatically generate pac[illegible] \"use them\" -- to be define[illegible] for both [illegible] combinations; also, docum[illegible]tation [illegible] this stage is the design ph[illegible]	[illegible]	[illegible]	Team	Saul (Dexuan)	M1, Design
OE Autobuilder rename		1	Done	Beth	Beth	M1, Design
Strip out LSB, non-LSB build work	Remove the LSB, non-LSB [illegible] and re-incorporate it with sstate	1	Done	Beth	Beth	M1, Design - one minor problem

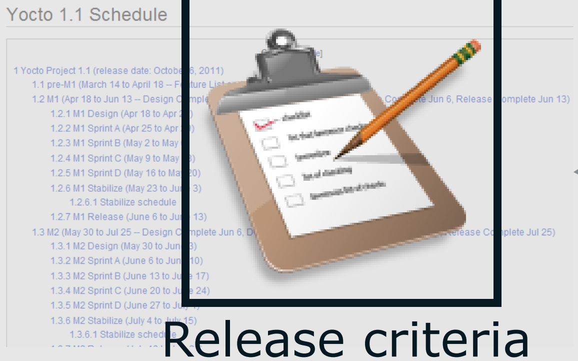
M1 Sprint A ([illegible] to Apr 29)

Feature Name	Description	Priority	Status	Source	Owner	Comments / Bugzilla Links
OE-Core	Restructuring, renaming, rebranding	1	Done	RP Notes	Richard	M1, Sprint A - only remaining piece is for OE Core to work without setting a distro (wasn't originally scheduled) - this is a new task for EO this week
SDK support in sanity test framework	This task includes enabling unfs and toolchain testing in sanity test framework, enabling toolchain testing on PRC autobuilder	1	Done	QA	Jiajun/Meilei	M1, Sprint A
User Creation at preinstall - status check	Design status check	1	Done	RP Notes	Mark (ScottG)	M1, Sprint A architecture was sent out as RFC
Check SRCREV in recipe files	should work, may need dev	2	Done	RP Notes	Richard/Ke	M1, Sprint A
Refactor BSPs to use topic branches	crownbay and fish river island BSP need to be changed to make use of the new eg20t/emgd/gma500 topic branches	2	Done	Tom	Tom	M1, Sprint A When other branches (ex. FRI 2) are added, they will use topic branches.
Retrospective	Hold a retrospective to discuss what went well and what can be improved in 1.1 with the meta-data team.	1	Done	Beth	Beth	M1, Sprint A

M1 Sprint B (May 2 to May 6)

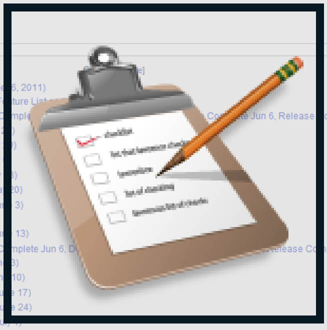
Breakdown of larger tasks

Create and track schedule



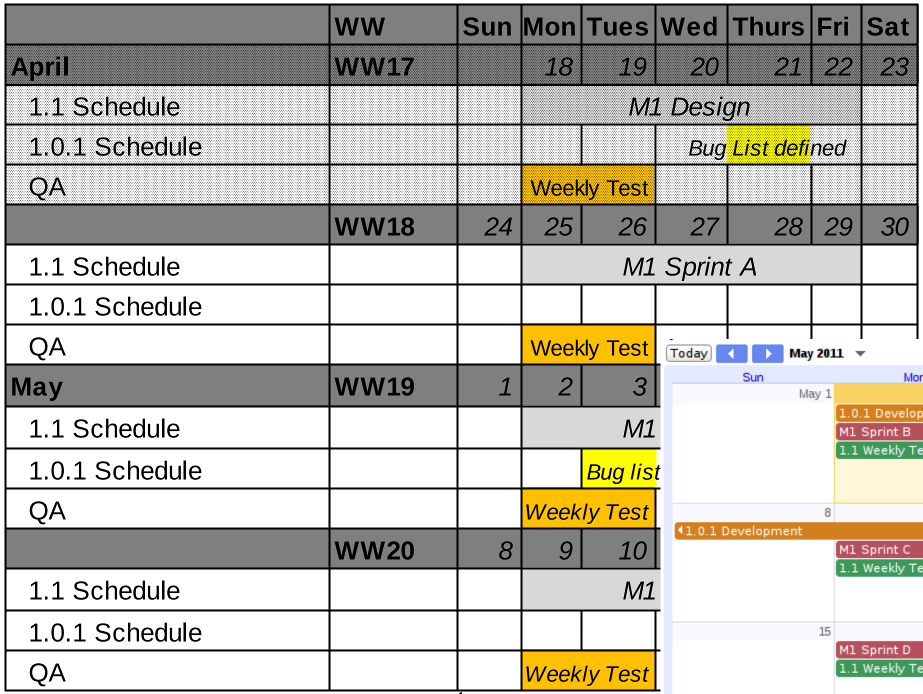
Yocto 1.1 Schedule

- 1 Yocto Project 1.1 (release date: Octobe[illegible]6, 2011)
  - 1.1 pre-M1 (March 14 to April 18 -- F[illegible]ture List[illegible]
  - 1.2 M1 (Apr 18 to Jun 13 -- Design C[illegible]mple[illegible] Co[illegible]te Jun 6, Release [illegible]omplete Jun 13)
    - 1.2.1 M1 Design (Apr 18 to Apr 2[illegible])
    - 1.2.2 M1 Sprint A (Apr 25 to Apr [illegible])
    - 1.2.3 M1 Sprint B (May 2 to May [illegible]
    - 1.2.4 M1 Sprint C (May 9 to May [illegible])
    - 1.2.5 M1 Sprint D (May 16 to Ma[illegible]20)
    - 1.2.6 M1 Stabilize (May 23 to Ju[illegible] 3)
      - 1.2.6.1 Stabilize schedule
    - 1.2.7 M1 Release (June 6 to Ju[illegible] 13)
  - 1.3 M2 (May 30 to Jul 25 -- Design C[illegible]mplete Jun 6, D[illegible] [illegible]elease Co[illegible]plete Jul 25)
    - 1.3.1 M2 Design (May 30 to Jun[illegible]3)
    - 1.3.2 M2 Sprint A (June 6 to Jun[illegible]10)
    - 1.3.3 M2 Sprint B (June 13 to Ju[illegible]e 17)
    - 1.3.4 M2 Sprint C (June 20 to Ju[illegible]e 24)
    - 1.3.5 M2 Sprint D (June 27 to Ju[illegible]
    - 1.3.6 M2 Stabilize (July 4 to July 15)
      - 1.3.6.1 Stabilize sched[illegible]



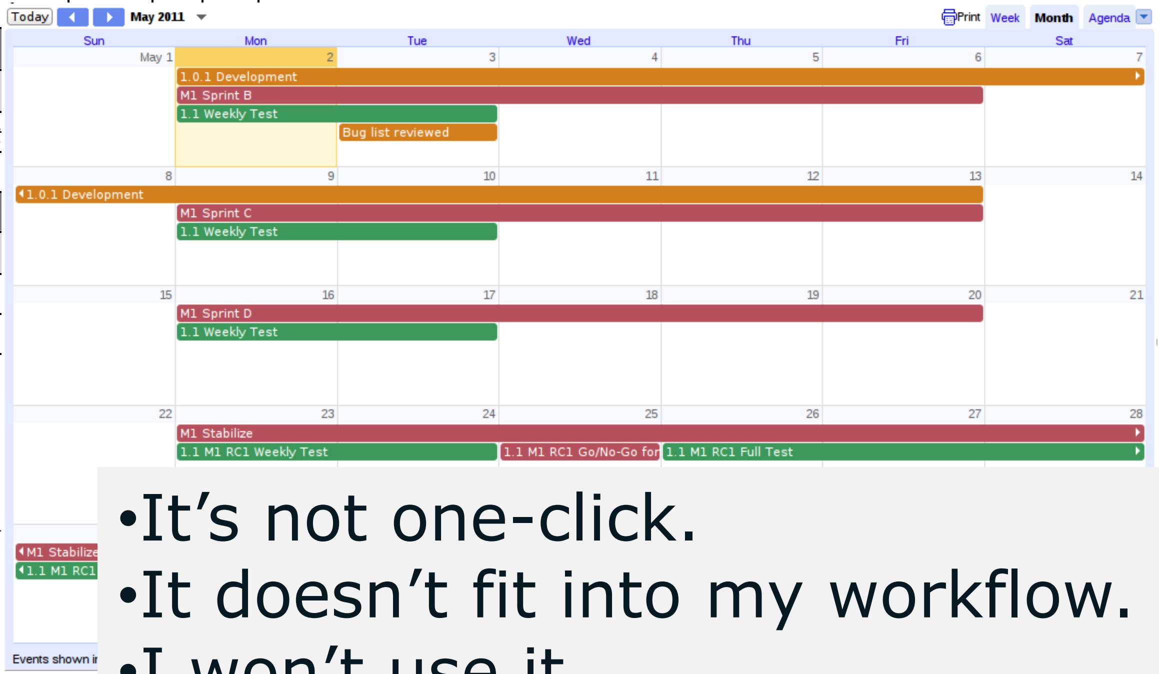
Release criteria

When I didn’t stick with what is familiar…



	WW	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
April	WW17		18	19	20	21	22	23
1.1 Schedule			M1 Design					
1.0.1 Schedule					Bug List defined			
QA			Weekly Test					
	WW18	24	25	26	27	28	29	30
1.1 Schedule			M1 Sprint A					
1.0.1 Schedule								
QA			Weekly Test					
May	WW19	1	2	3				
1.1 Schedule			M1					
1.0.1 Schedule				Bug list				
QA			Weekly Test					
	WW20	8	9	10				
1.1 Schedule			M1					
1.0.1 Schedule								
QA			Weekly Test					

versus



Today May 2011 Print Week Month Agenda

Sun Mon Tue Wed Thu Fri Sat

May 1 2 3 4 5 6 7
1.0.1 Development
M1 Sprint B
1.1 Weekly Test
Bug list reviewed

8 9 10 11 12 13 14
1.0.1 Development
M1 Sprint C
1.1 Weekly Test

15 16 17 18 19 20 21
M1 Sprint D
1.1 Weekly Test

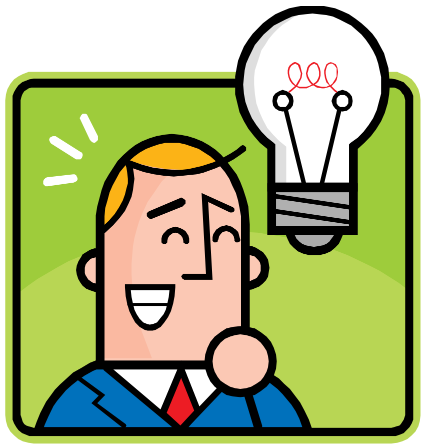
22 23 24 25 26 27 28
M1 Stabilize
1.1 M1 RC1 Weekly Test
1.1 M1 RC1 Go/No-Go for
1.1 M1 RC1 Full Test

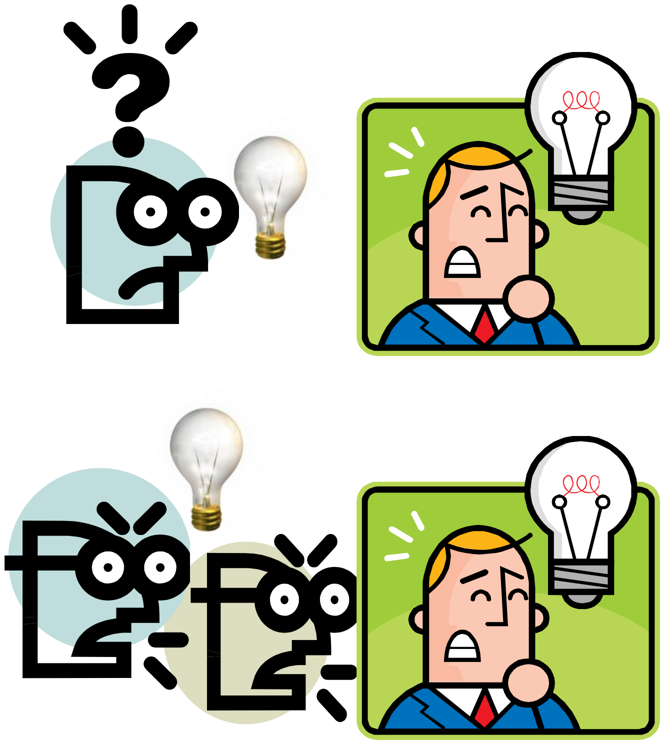
M1 Stabilize
1.1 M1 RC1

Events shown i

- It’s not one-click.
- It doesn’t fit into my workflow.
- I won’t use it.
- It doesn’t meet my needs.

Tip #3: Stick with what is familiar





When you don't…

At best, your message is misunderstood, and you feel undervalued.

At worst, your message is offensive, leaving you and the team feeling undervalued.

Better to leverage what exists and move towards new solutions together.

The Tips on Delivering Quality

- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

My first project: Create a Product Requirements Document

But, …

- Project was over halfway done.
- Features and requirements were already nailed down on the Wiki.
- People would not attend meetings to discuss or follow up on action items from meetings.

The team didn't need a PRD.

The team needed to convey to other leaders that we had a stable, mature process.

Tip #4: Focus on the need, not the name

This statement…	**… becomes THIS:**
Where is our Product Requirements Document?	How do we track our feature ideas so we know what we want to implement and by when?
Where is our GANTT chart?	How do we know what we will deliver and by when?
Where are our Release Criteria?	How do we know if a release is “good enough” to be labeled as “stable?”

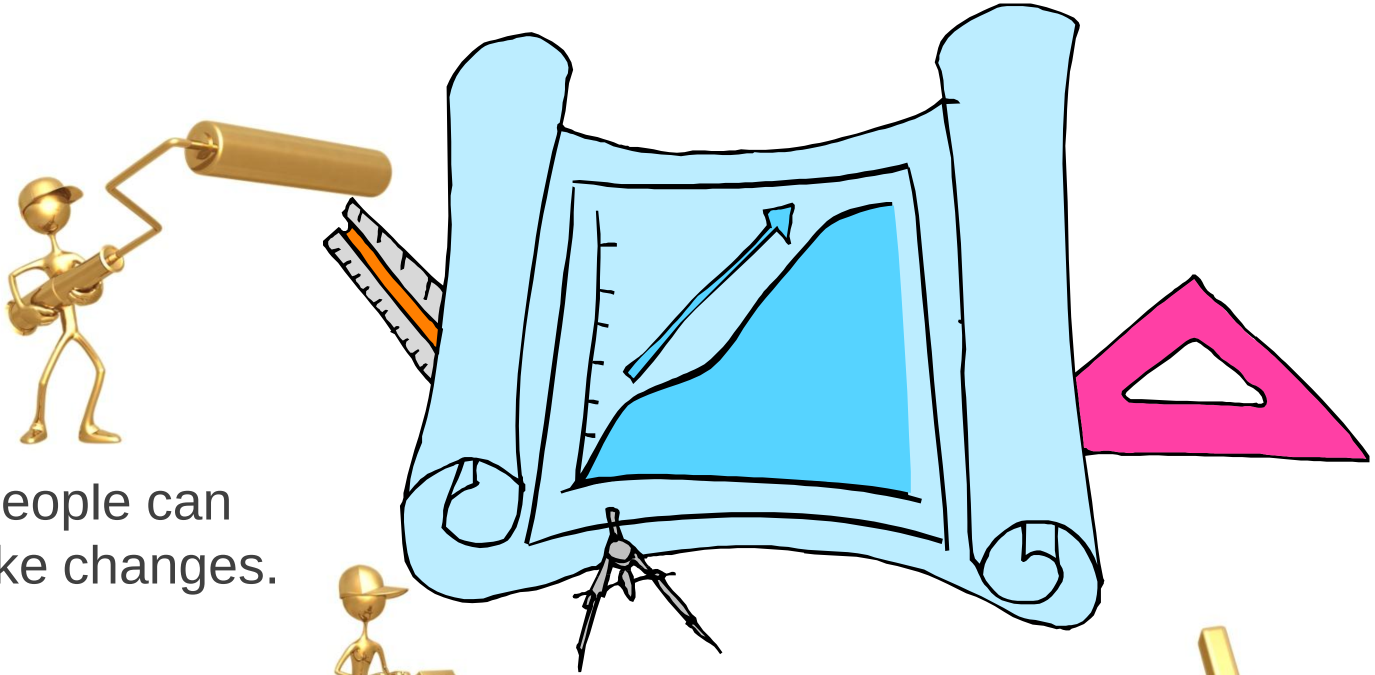
The Tips on Delivering Quality

- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

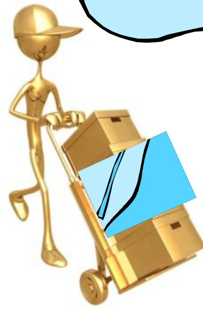
Remember “Stone Soup?”

Eventually, the whole village was fed…

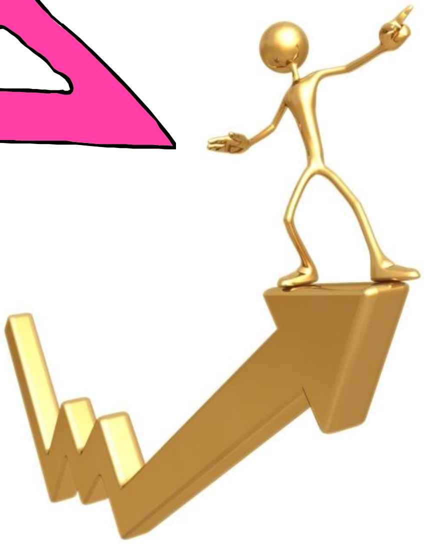
Tip #5: Begin all proposals with a strawman



People can make changes.



People can add or remove pieces.

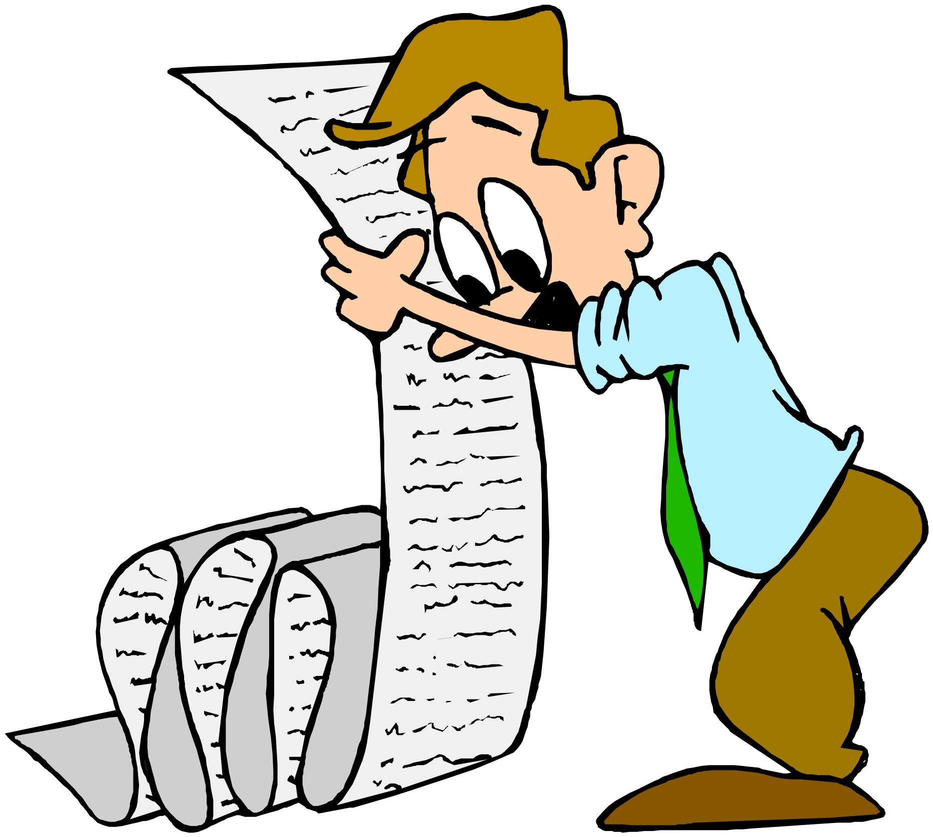


You retain influence in the overall direction and ensure momentum is maintained.

The Tips on Delivering Quality

- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

Tip #6: Always have a list of tasks needing volunteers



- This could be:
  - A bug database.
  - A Janitor’s List on Wiki or website.
  - A “Features Not Scheduled” List on Wiki or website.
  - A 1-1 with key maintainers or leaders in the project.
- Include scope of task, time required, technical expertise required.

INTEL OPEN SOURCE TECHNOLOGY CENTER

Getting newcomers actively involved…

- …Provides them with the skills and experience to make bigger contributions

And

- Makes them feel part of the community

INTEL OPEN SOURCE TECHNOLOGY CENTER

The Tips on Delivering Quality

- Tip #1: Leverage the open source environment
- Tip #2: Leverage the open source community
- Tip #3: Stick with what is familiar
- Tip #4: Focus on the need, not the name
- Tip #5: Begin all proposals with a strawman
- Tip #6: Always have a list of tasks needing volunteers
- Tip #7: Respect your community

Tip #7: Respect your community

Acknowledge contributions

- Even when they aren't what you are looking for.
- Especially when they are.

Embrace diversity

- Different viewpoints, agendas, skill sets and backgrounds enhance your program.
- Serendipity favors variety.

Actively listen to feedback

- Document and follow up on input.

Celebrate your milestones

- Even when everyone is virtual, find a way to celebrate the milestones.

INTEL OPEN SOURCE
TECHNOLOGY CENTER