AALBORG UNIVERSITY

Scheduling and Rationality

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
E2-201
Information Systems
March 8, 2007

Scheduling & Tracking

Why Are Projects Late?

- An unrealistic deadline established by outsiders
- Changing customer requirements not reflected in schedule changes
- An honest underestimate of the effort required to do the job
- Predictable and/or unpredictable risks that were not considered at project start
- Technical difficulties that were not foreseen

Pressman 2000

Why Are Projects Late?

Human difficulties that were not foreseen

Miscommunication among project staff

A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem

Scheduling Principles

Compartmentalization—define distinct tasks

Interdependency—indicate task interrelationships

Effort validation—be sure resources are available

Defined responsibilities—people must be assigned

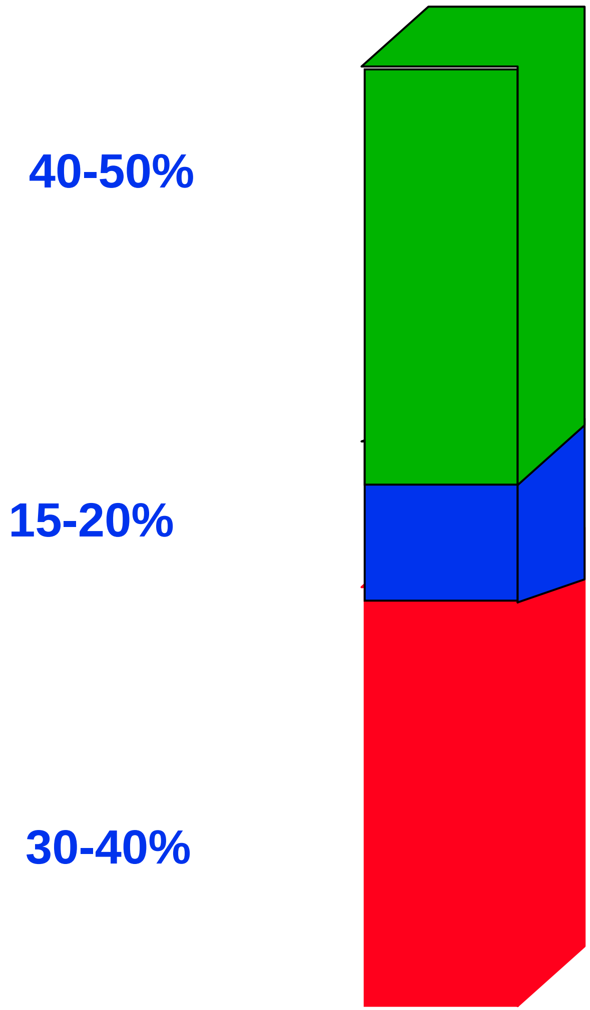
Defined outcomes—each task must have an output

Defined milestones—review for quality

Pressman 2000

Pressman 2000

Effort Allocation



"front end" activities

- customer communication
- analysis
- design
- review and modification

construction activities

- coding or code generation

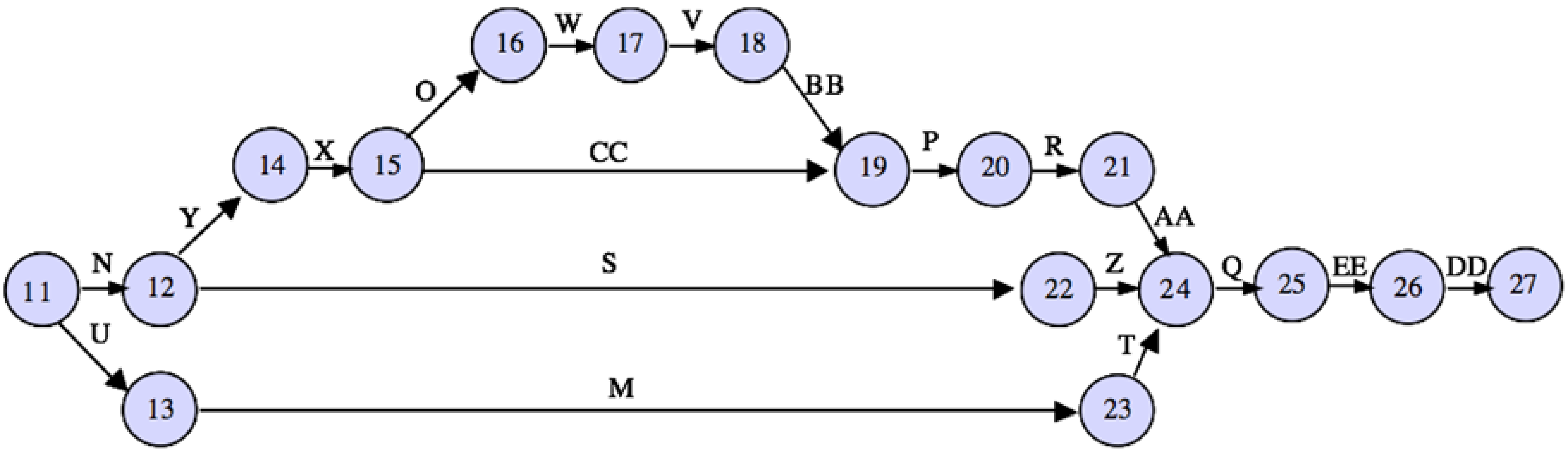
testing and installation

- unit, integration
- white-box, black box
- regression

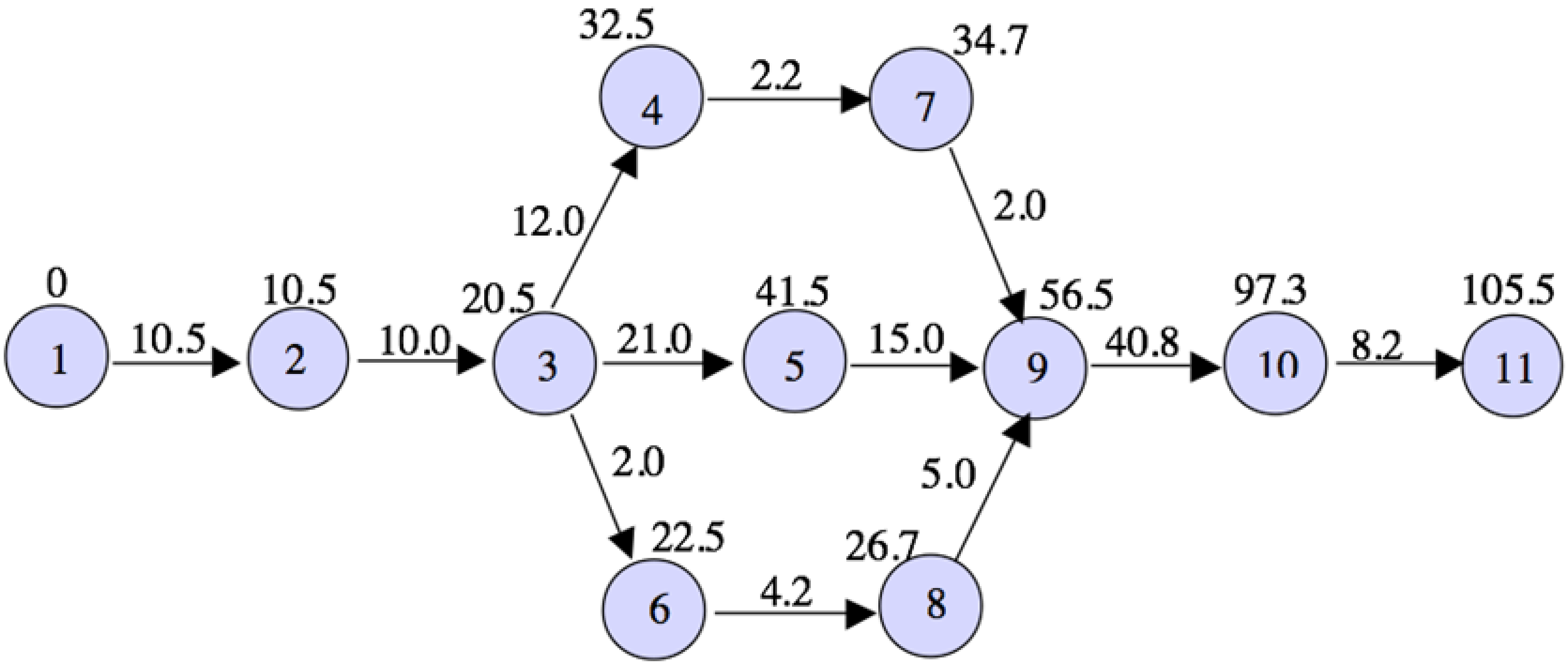
Questions Addressed by Scheduling

Completion date?
On Schedule?
Within Budget?
Critical Activities?
How can the project be finished early at the least cost?

PERT Project Network



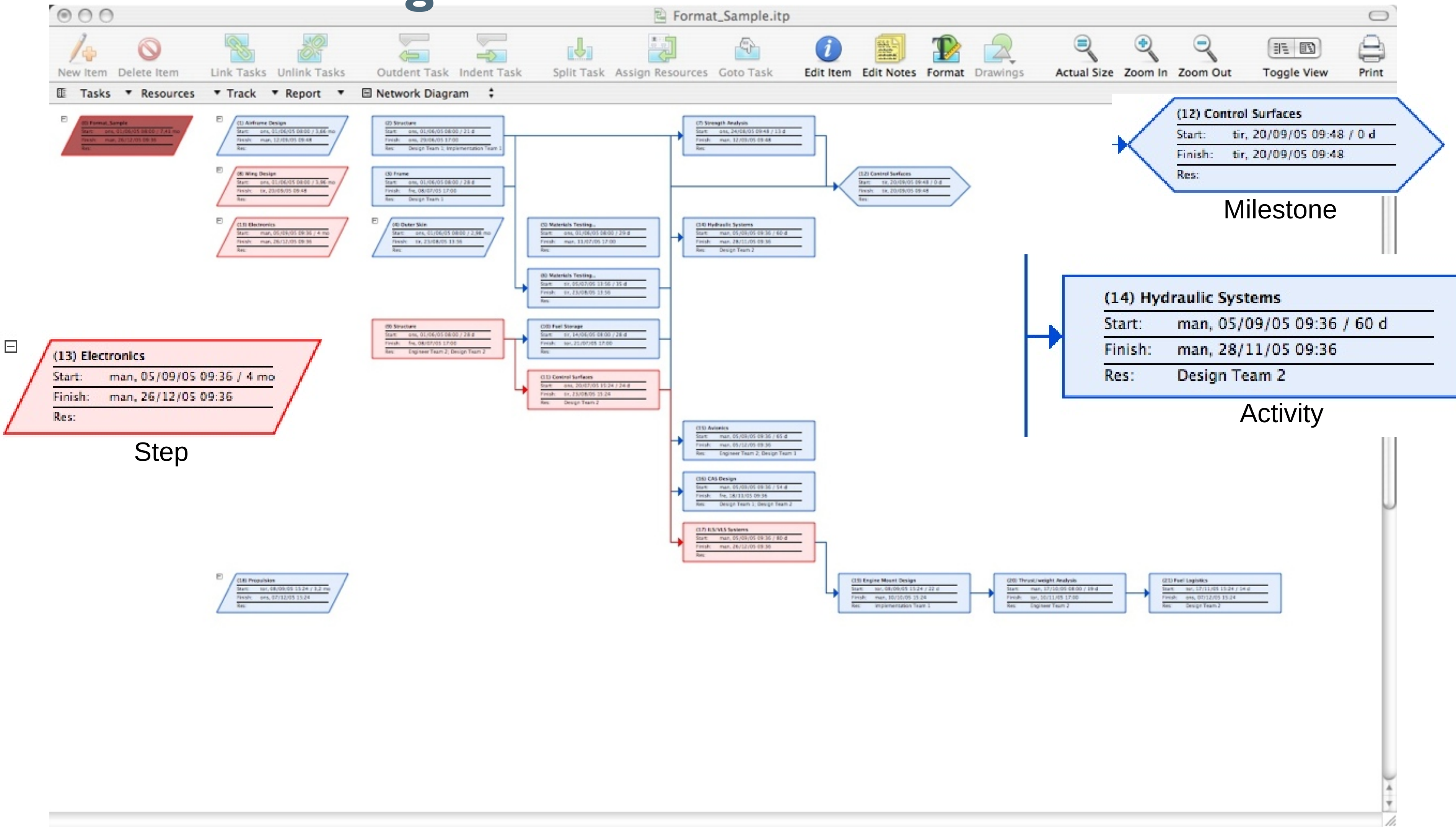
Pert Chart with Milestone Time Label



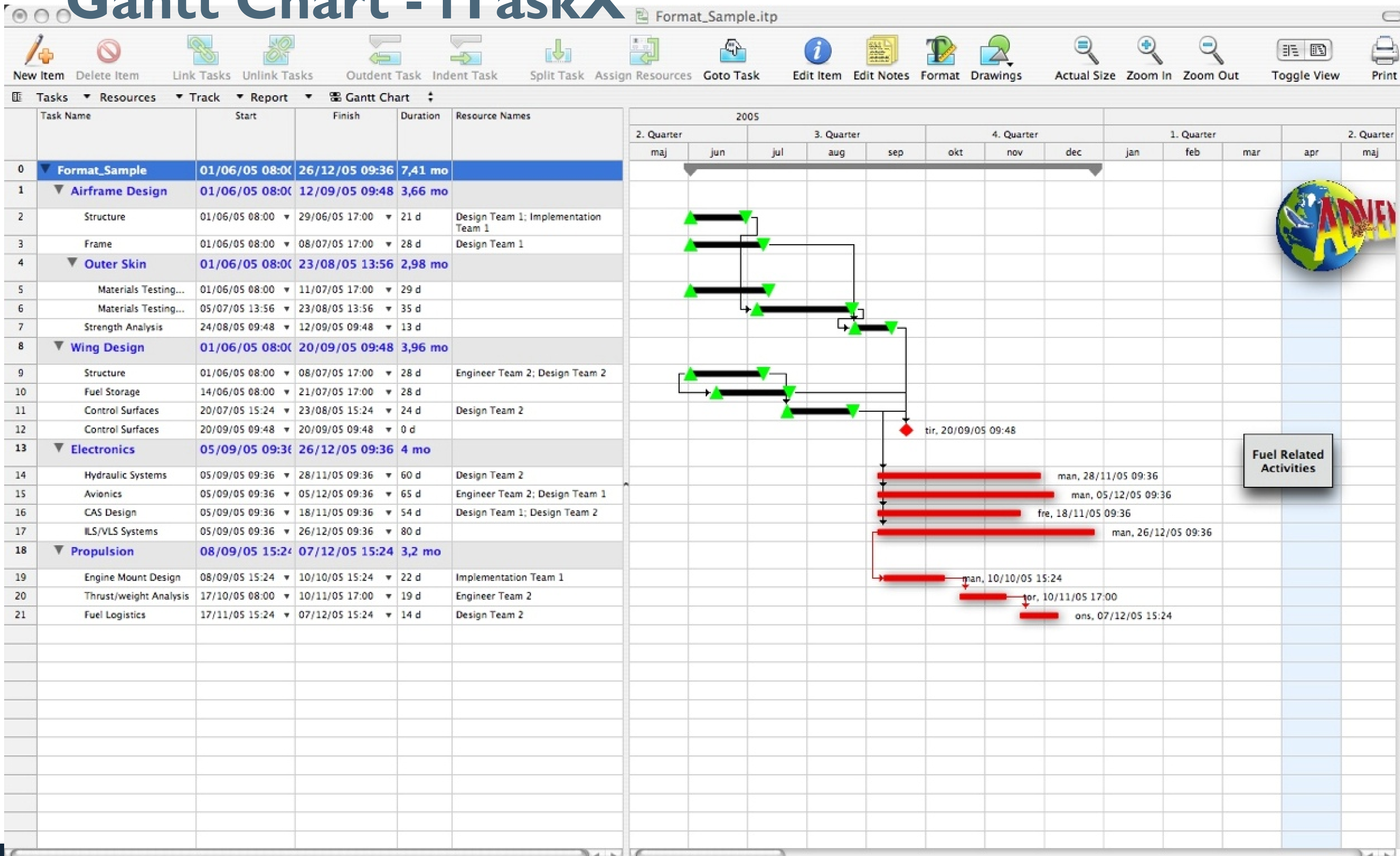
Activity Scheduling

Earliest start time
Earliest finish time
Latest start time
Latest finish time

Network diagram - iTaskX



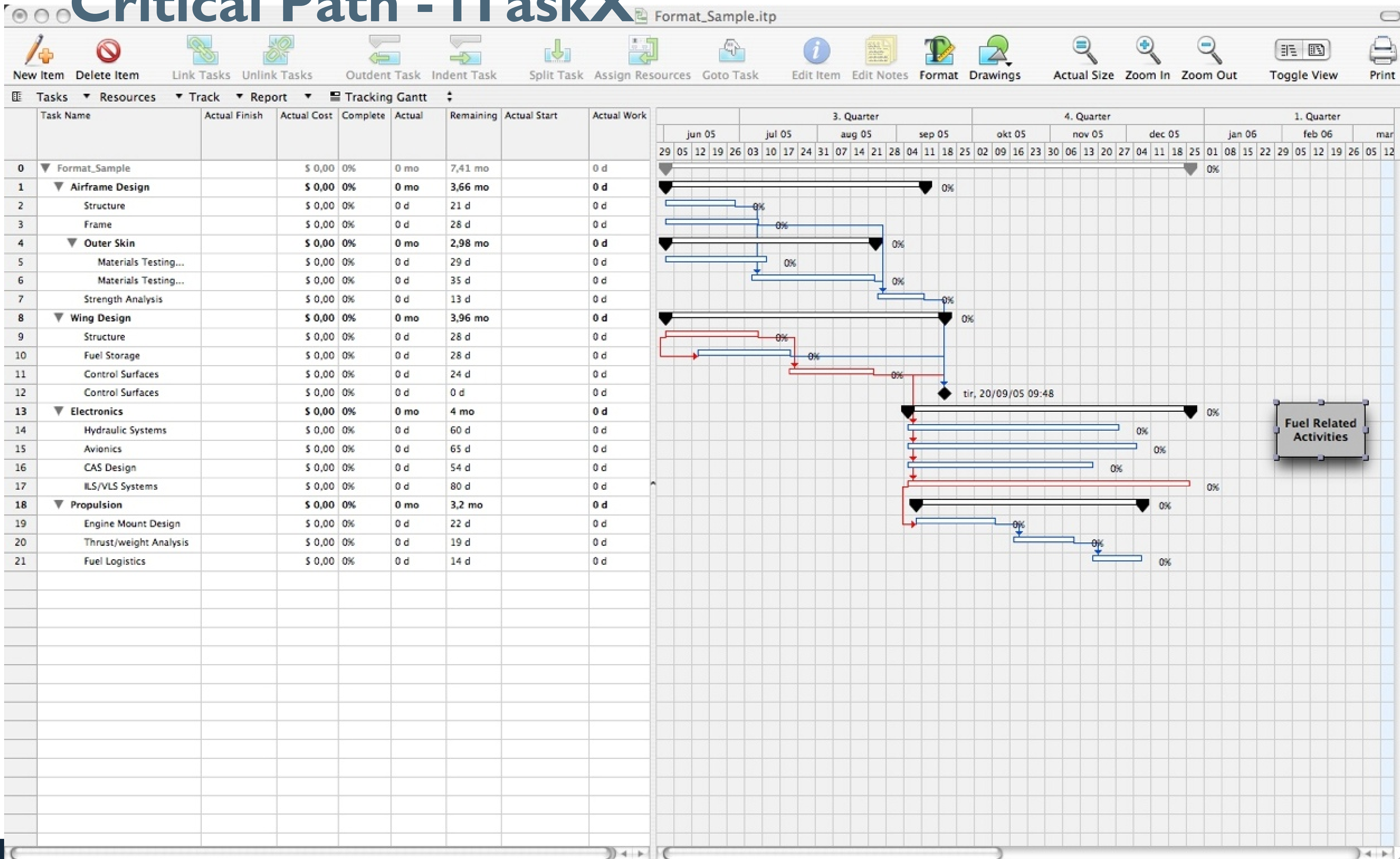
Gantt Chart - iTaskX



Critical Path

Longest path through a network
Minimum project completion time

Critical Path - iTaskX

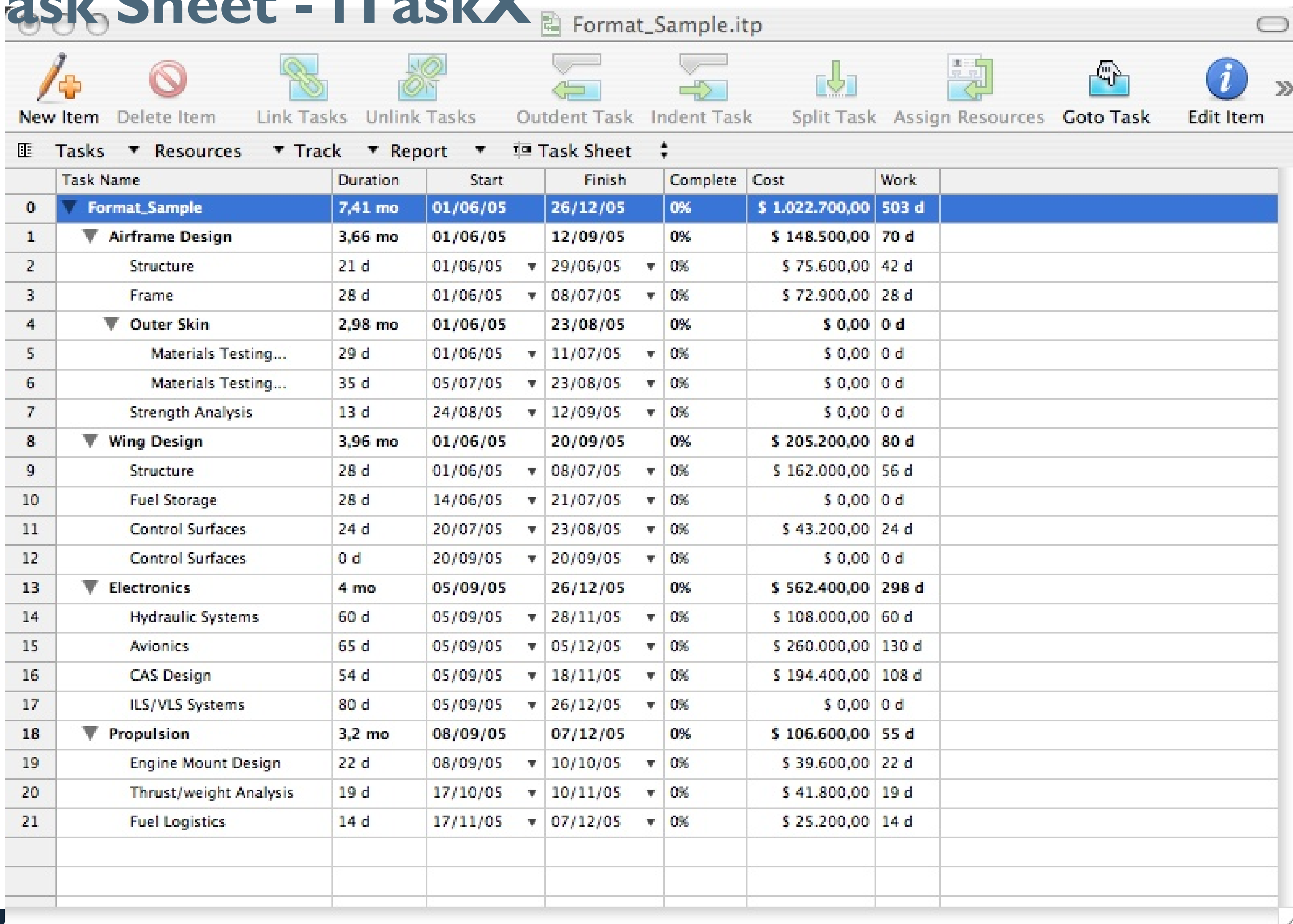


	Task Name	Actual Finish	Actual Cost	Complete	Actual	Remaining	Actual Start	Actual Work
0	Format_Sample		$ 0,00	0%	0 mo	7,41 mo		0 d
1	Airframe Design		$ 0,00	0%	0 mo	3,66 mo		0 d
2	Structure		$ 0,00	0%	0 d	21 d		0 d
3	Frame		$ 0,00	0%	0 d	28 d		0 d
4	Outer Skin		$ 0,00	0%	0 mo	2,98 mo		0 d
5	Materials Testing...		$ 0,00	0%	0 d	29 d		0 d
6	Materials Testing...		$ 0,00	0%	0 d	35 d		0 d
7	Strength Analysis		$ 0,00	0%	0 d	13 d		0 d
8	Wing Design		$ 0,00	0%	0 mo	3,96 mo		0 d
9	Structure		$ 0,00	0%	0 d	28 d		0 d
10	Fuel Storage		$ 0,00	0%	0 d	28 d		0 d
11	Control Surfaces		$ 0,00	0%	0 d	24 d		0 d
12	Control Surfaces		$ 0,00	0%	0 d	0 d		0 d
13	Electronics		$ 0,00	0%	0 mo	4 mo		0 d
14	Hydraulic Systems		$ 0,00	0%	0 d	60 d		0 d
15	Avionics		$ 0,00	0%	0 d	65 d		0 d
16	CAS Design		$ 0,00	0%	0 d	54 d		0 d
17	ILS/VLS Systems		$ 0,00	0%	0 d	80 d		0 d
18	Propulsion		$ 0,00	0%	0 mo	3,2 mo		0 d
19	Engine Mount Design		$ 0,00	0%	0 d	22 d		0 d
20	Thrust/weight Analysis		$ 0,00	0%	0 d	19 d		0 d
21	Fuel Logistics		$ 0,00	0%	0 d	14 d		0 d

Activity Slack

Definition: Slack is the amount of time an activity can be delayed without delaying the project

Task Sheet - iTaskX



	Task Name	Duration	Start	Finish	Complete	Cost	Work
0	**Format_Sample**	**7,41 mo**	**01/06/05**	**26/12/05**	**0%**	**$ 1.022.700,00**	**503 d**
1	**Airframe Design**	**3,66 mo**	**01/06/05**	**12/09/05**	**0%**	**$ 148.500,00**	**70 d**
2	Structure	21 d	01/06/05	29/06/05	0%	$ 75.600,00	42 d
3	Frame	28 d	01/06/05	08/07/05	0%	$ 72.900,00	28 d
4	**Outer Skin**	**2,98 mo**	**01/06/05**	**23/08/05**	**0%**	**$ 0,00**	**0 d**
5	Materials Testing...	29 d	01/06/05	11/07/05	0%	$ 0,00	0 d
6	Materials Testing...	35 d	05/07/05	23/08/05	0%	$ 0,00	0 d
7	Strength Analysis	13 d	24/08/05	12/09/05	0%	$ 0,00	0 d
8	**Wing Design**	**3,96 mo**	**01/06/05**	**20/09/05**	**0%**	**$ 205.200,00**	**80 d**
9	Structure	28 d	01/06/05	08/07/05	0%	$ 162.000,00	56 d
10	Fuel Storage	28 d	14/06/05	21/07/05	0%	$ 0,00	0 d
11	Control Surfaces	24 d	20/07/05	23/08/05	0%	$ 43.200,00	24 d
12	Control Surfaces	0 d	20/09/05	20/09/05	0%	$ 0,00	0 d
13	**Electronics**	**4 mo**	**05/09/05**	**26/12/05**	**0%**	**$ 562.400,00**	**298 d**
14	Hydraulic Systems	60 d	05/09/05	28/11/05	0%	$ 108.000,00	60 d
15	Avionics	65 d	05/09/05	05/12/05	0%	$ 260.000,00	130 d
16	CAS Design	54 d	05/09/05	18/11/05	0%	$ 194.400,00	108 d
17	ILS/VLS Systems	80 d	05/09/05	26/12/05	0%	$ 0,00	0 d
18	**Propulsion**	**3,2 mo**	**08/09/05**	**07/12/05**	**0%**	**$ 106.600,00**	**55 d**
19	Engine Mount Design	22 d	08/09/05	10/10/05	0%	$ 39.600,00	22 d
20	Thrust/weight Analysis	19 d	17/10/05	10/11/05	0%	$ 41.800,00	19 d
21	Fuel Logistics	14 d	17/11/05	07/12/05	0%	$ 25.200,00	14 d

Task Usage - iTaskX

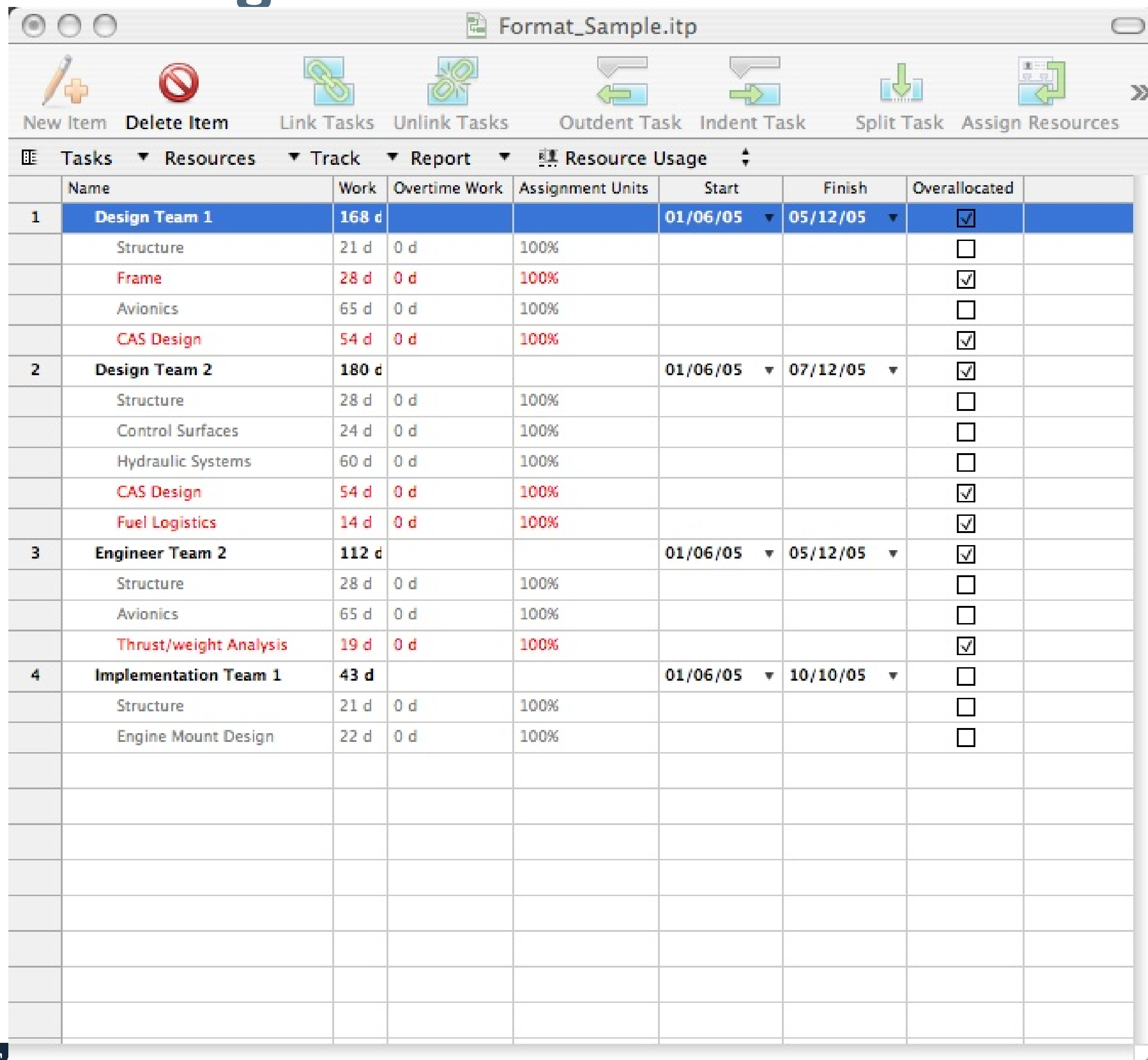
Format_Sample.itp

New Item | Delete Item | Link Tasks | Unlink Tasks | Outdent Task | Indent Task | Split Task | Assign Resources

Tasks ▾ Resources ▾ Track ▾ Report ▾ Task Usage

	Name	Work	Overtime Work	Duration	Assignment Units	Start	Finish	Overallocated
0	**Format_Sample**	**503 d**	**0 d**	**7,41 mo**		**01/06/05**	**26/12/05**	[ ]
1	**Airframe Design**	**70 d**	**0 d**	**3,66 mo**		**01/06/05**	**12/09/05**	[ ]
2	Structure	42 d	0 d	21 d		01/06/05	29/06/05	[ ]
	Design Team 1	21 d	0 d		100%			[ ]
	Implementation	21 d	0 d		100%			[ ]
3	Frame	28 d	0 d	28 d		01/06/05	08/07/05	[x]
	Design Team 1	28 d	0 d		100%			[x]
4	**Outer Skin**	**0 d**	**0 d**	**2,98 mo**		**01/06/05**	**23/08/05**	[ ]
5	Materials Testing...	0 d	0 d	29 d		01/06/05	11/07/05	[ ]
6	Materials Testing...	0 d	0 d	35 d		05/07/05	23/08/05	[ ]
7	Strength Analysis	0 d	0 d	13 d		24/08/05	12/09/05	[ ]
8	**Wing Design**	**80 d**	**0 d**	**3,96 mo**		**01/06/05**	**20/09/05**	[ ]
9	Structure	56 d	0 d	28 d		01/06/05	08/07/05	[ ]
	Engineer Team 2	28 d	0 d		100%			[ ]
	Design Team 2	28 d	0 d		100%			[ ]
10	Fuel Storage	0 d	0 d	28 d		14/06/05	21/07/05	[ ]
11	Control Surfaces	24 d	0 d	24 d		20/07/05	23/08/05	[ ]
	Design Team 2	24 d	0 d		100%			[ ]
12	Control Surfaces	0 d	0 d	0 d		20/09/05	20/09/05	[ ]
13	**Electronics**	**298 d**	**0 d**	**4 mo**		**05/09/05**	**26/12/05**	[ ]
18	**Propulsion**	**55 d**	**0 d**	**3,2 mo**		**08/09/05**	**07/12/05**	[ ]
19	Engine Mount Design	22 d	0 d	22 d		08/09/05	10/10/05	[ ]
	Implementation	22 d	0 d		100%			[ ]
20	Thrust/weight Analysis	19 d	0 d	19 d		17/10/05	10/11/05	[x]
	Engineer Team 2	19 d	0 d		100%			[x]
21	Fuel Logistics	14 d	0 d	14 d		17/11/05	07/12/05	[x]
	Design Team 2	14 d	0 d		100%			[x]

Resource Usage - iTaskX



	Name	Work	Overtime Work	Assignment Units	Start	Finish	Overallocated
1	Design Team 1	168 d			01/06/05	05/12/05	☑
	Structure	21 d	0 d	100%			☐
	Frame	28 d	0 d	100%			☑
	Avionics	65 d	0 d	100%			☐
	CAS Design	54 d	0 d	100%			☑
2	Design Team 2	180 d			01/06/05	07/12/05	☑
	Structure	28 d	0 d	100%			☐
	Control Surfaces	24 d	0 d	100%			☐
	Hydraulic Systems	60 d	0 d	100%			☐
	CAS Design	54 d	0 d	100%			☑
	Fuel Logistics	14 d	0 d	100%			☑
3	Engineer Team 2	112 d			01/06/05	05/12/05	☑
	Structure	28 d	0 d	100%			☐
	Avionics	65 d	0 d	100%			☐
	Thrust/weight Analysis	19 d	0 d	100%			☑
4	Implementation Team 1	43 d			01/06/05	10/10/05	☐
	Structure	21 d	0 d	100%			☐
	Engine Mount Design	22 d	0 d	100%			☐

Cost Table - iTaskX

Format_Sample.itp

New Item | Delete Item | Link Tasks | Unlink Tasks | Outdent Task | Indent Task | Split Task | Assign Resources

Tasks | Resources | Track | Report | Resource Usage

	Name	Cost	Baseline Cost	Cost Variance	Actual Cost	Remaining Cost
1	**Design Team 1**	**$ 32**	**$ 310.050,00**	**$ 14.850,00**	**$ 0,00**	**$ 324.900,00**
	Structure	$ 37.	$ 37.800,00	$ 0,00	$ 0,00	$ 37.800,00
	Frame	$ 72.	$ 58.050,00	$ 14.850,00	$ 0,00	$ 72.900,00
	Avionics	$ 11	$ 117.000,00	$ 0,00	$ 0,00	$ 117.000,00
	CAS Design	$ 97.	$ 97.200,00	$ 0,00	$ 0,00	$ 97.200,00
2	**Design Team 2**	**$ 34**	**$ 345.825,00**	**$ 675,00**	**$ 0,00**	**$ 346.500,00**
	Structure	$ 72.	$ 72.225,00	$ 675,00	$ 0,00	$ 72.900,00
	Control Surfaces	$ 43.	$ 43.200,00	$ 0,00	$ 0,00	$ 43.200,00
	Hydraulic Systems	$ 10	$ 108.000,00	$ 0,00	$ 0,00	$ 108.000,00
	CAS Design	$ 97.	$ 97.200,00	$ 0,00	$ 0,00	$ 97.200,00
	Fuel Logistics	$ 25.	$ 25.200,00	$ 0,00	$ 0,00	$ 25.200,00
3	**Engineer Team 2**	**$ 27**	**$ 273.075,00**	**$ 825,00**	**$ 0,00**	**$ 273.900,00**
	Structure	$ 89.	$ 88.275,00	$ 825,00	$ 0,00	$ 89.100,00
	Avionics	$ 14	$ 143.000,00	$ 0,00	$ 0,00	$ 143.000,00
	Thrust/weight Analysis	$ 41.	$ 41.800,00	$ 0,00	$ 0,00	$ 41.800,00
4	**Implementation Team 1**	**$ 77.**	**$ 77.400,00**	**$ 0,00**	**$ 0,00**	**$ 77.400,00**
	Structure	$ 37.	$ 37.800,00	$ 0,00	$ 0,00	$ 37.800,00
	Engine Mount Design	$ 39.	$ 39.600,00	$ 0,00	$ 0,00	$ 39.600,00

Scheduling Methods: Strengths

These methods are useful prior to and during a project

They are straightforward in concept and are supported by software

Graphical representation of the project's tasks help to show the task interrelationships

Highlighting the project's critical path and task slack time allows to focus on critical aspects of project-time, costs and people

Project management software usually provides excellent project tracking documentation

These methods are applicable in a wide variety of projects.

Pressman 2001

Scheduling Methods: Weaknesses

Project tasks have to be clearly defined as well as their relationships to each other

Do not deal very well with task overlap. They assume the following tasks begin after their preceding tasks end

They are only as good as the time estimates

By design, the project manager will normally focus more attention on the critical path tasks than other tasks, which could be problematic for near-critical path tasks if overlooked

Pressman 2001

Tracking: Elementary Metrics

Unit of measure	Characteristics addressed
Counts of physical source lines of code	Size, progress, reuse
Counts of staff-hours expended	Effort, cost, resource allocations
Calendar dates	Schedule
Counts of software problems and defects	Quality, readiness for delivery, improvement trends

Tracking - Manpower & Effort

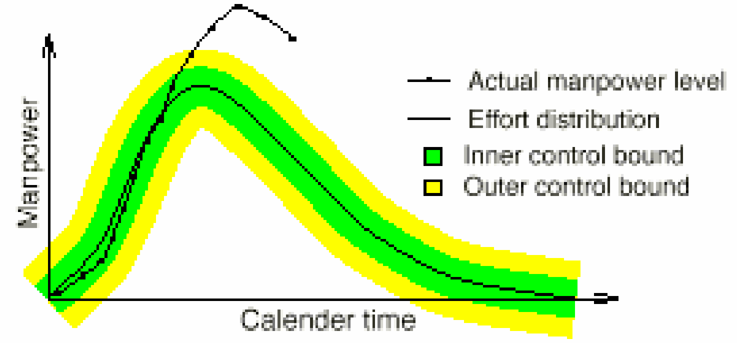


Figure 13: Rate curve. The actual effort values are plotted against the distribution.

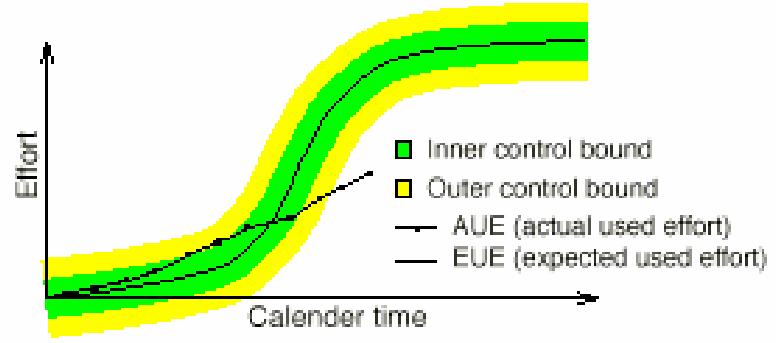


Figure 14: Cumulative curve. The actual effort values are plotted against the distribution.

Steen Andersen, Peter Stegenborg Larsen, Carsten Lindholst: *Evaluation and Evolution of Navi - a Web Based Tool for Project Planning and Tracking*, Masters Thesis, Computer Science, Aalborg University, 1998.

Tracking - Lines of Code & Defects

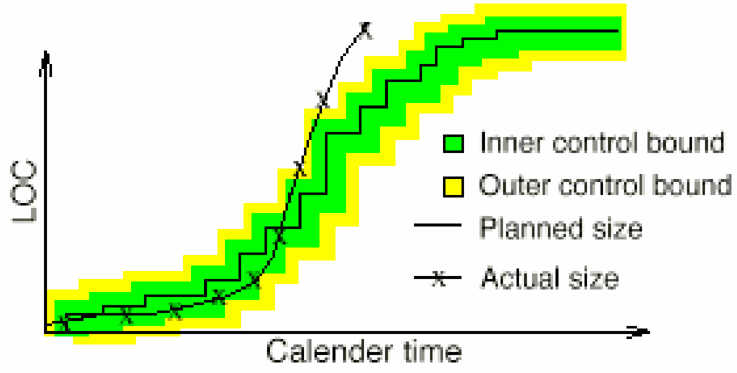


Figure 16: The actual size in LOC plotted against the planned size

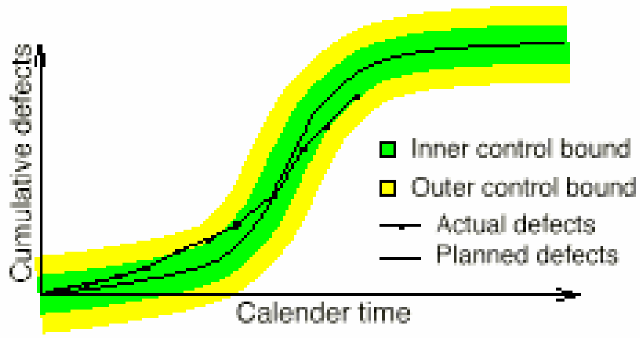
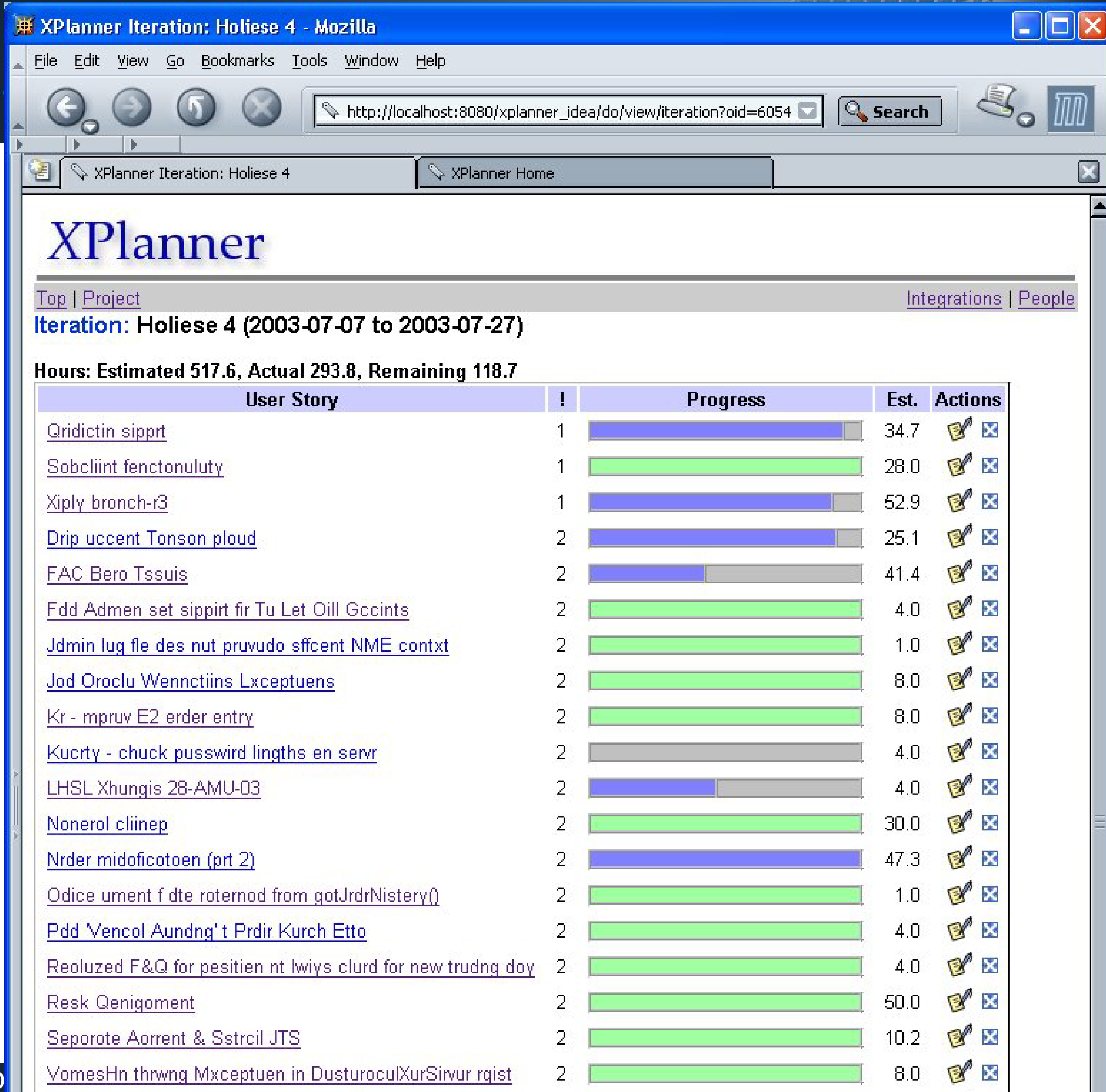


Figure 19: Tracking cumulative defect arrival against a planned Rayleigh distribution.

Steen Andersen, Peter Stegenborg Larsen, Carsten Lindholst: *Evaluation and Evolution of Navi - a Web Based Tool for Project Planning and Tracking*, Masters Thesis, Computer Science, Aalborg University, 1998.

XPlanner - www.xplanner.org/

AALBOR



XPlanner

Top | Project | Integrations | People

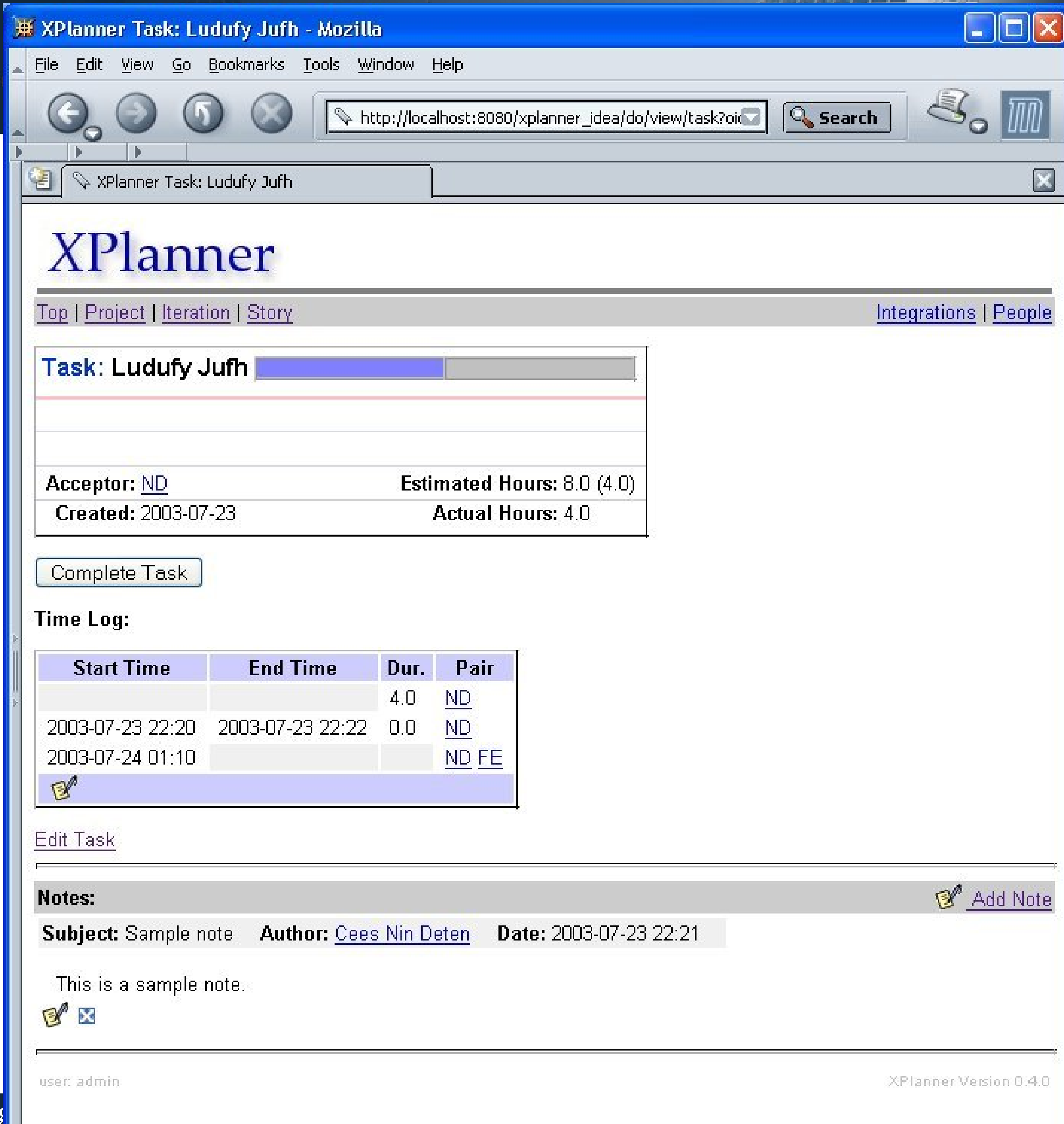
Iteration: Holiese 4 (2003-07-07 to 2003-07-27)

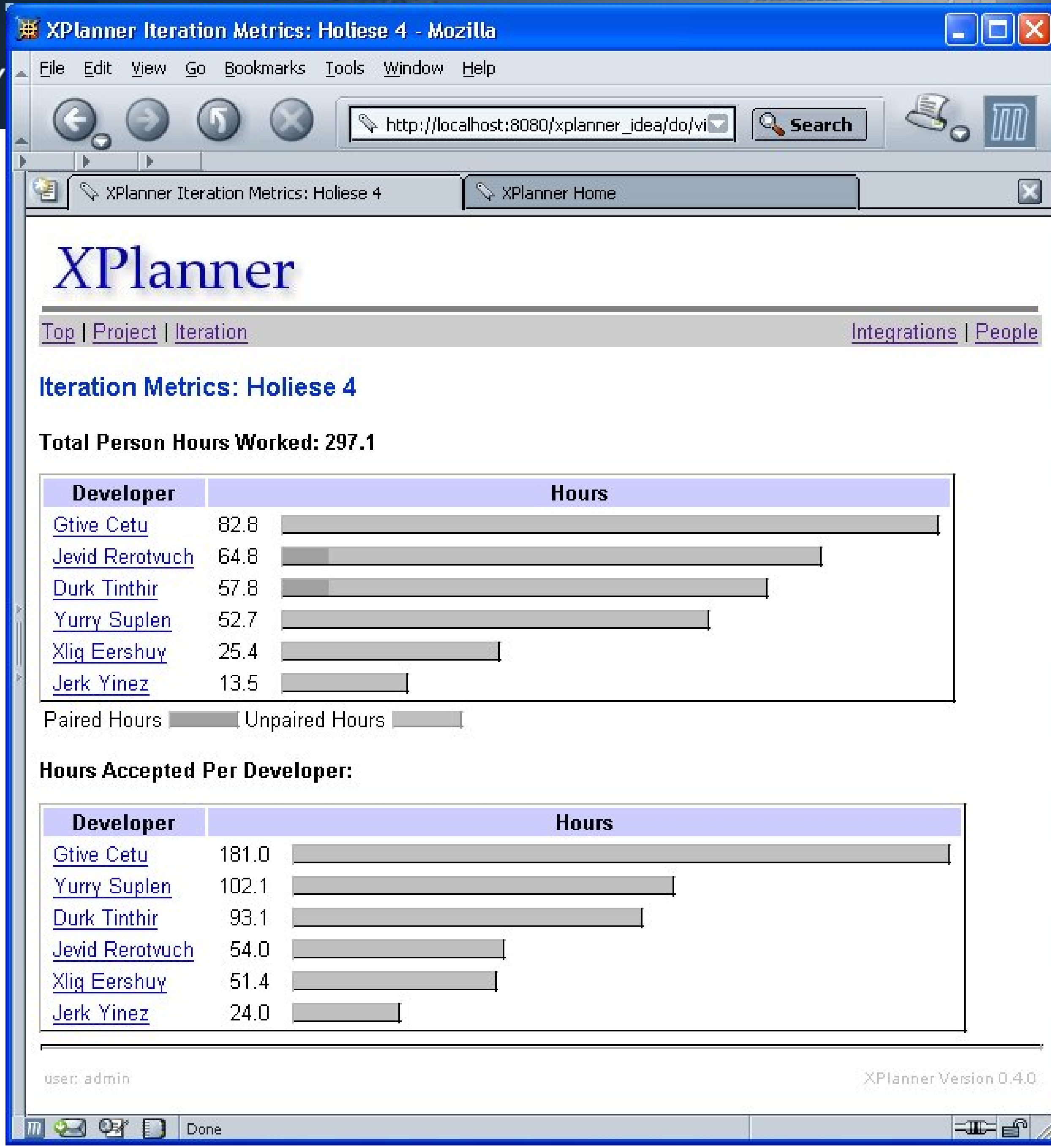
Hours: Estimated 517.6, Actual 293.8, Remaining 118.7

User Story	!	Progress	Est.	Actions
Qridictin sipprt	1		34.7	
Sobcliint fenctonuluty	1		28.0	
Xiply bronch-r3	1		52.9	
Drip uccent Tonson ploud	2		25.1	
FAC Bero Tssuis	2		41.4	
Fdd Admen set sippirt fir Tu Let Oill Gccints	2		4.0	
Jdmin lug fle des nut pruvudo sffcent NME contxt	2		1.0	
Jod Oroclu Wennctiins Lxceptuens	2		8.0	
Kr - mpruv E2 erder entry	2		8.0	
Kucrty - chuck pusswird lingths en servr	2		4.0	
LHSL Xhungis 28-AMU-03	2		4.0	
Nonerol cliinep	2		30.0	
Nrder midoficotoen (prt 2)	2		47.3	
Odice ument f dte roternod from gotJrdrNistery()	2		1.0	
Pdd 'Vencol Aundng' t Prdir Kurch Etto	2		4.0	
Reoluzed F&Q for pesitien nt lwiys clurd for new trudng doy	2		4.0	
Resk Qenigoment	2		50.0	
Seporote Aorrent & Sstrcil JTS	2		10.2	
VomesHn thrwng Mxceptuen in DusturoculXurSirvur rqist	2		8.0	

www.aau.dk


XPlanner Story: Lsr Erondly BrdorVds - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/xplanner_idea/do/view/userstor
Search
XPlanner Story: Lsr Erondly BrdorVds
XPlanner
Top | Project | Iteration
Integrations | People
Story: Lsr Erondly BrdorVds
http://example.com/design_notes.txt
Feature
Description
First Feature
This is the first feature
Another one
This is another feature
Item 1
Item 2
Priority: 4
Estimated Hours: 14.0
Actual Hours: 6.6
Task Name
Type
Progress
Est.
Acc.
Disposition
Actions
Lodofy sirvr
Ueture
3.0
ND
Dlinnd
Ludufy Jufh
Feature
8.0
ND
Planned
Oudefy bllng
Mitori
3.0
ND
Blunnid
Edit Story | Create Task
Notes:
Add Note
user: admin
XPlanner Version 0.4.0


XPlanner Task: Ludufy Jufh - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/xplanner_idea/do/view/task?oi
Search
XPlanner Task: Ludufy Jufh
XPlanner
Top | Project | Iteration | Story
Integrations | People
Task: Ludufy Jufh
Acceptor: ND
Estimated Hours: 8.0 (4.0)
Created: 2003-07-23
Actual Hours: 4.0
Complete Task
Time Log:
Start Time
End Time
Dur.
Pair
4.0
ND
2003-07-23 22:20
2003-07-23 22:22
0.0
ND
2003-07-24 01:10
ND FE
Edit Task
Notes:
Add Note
Subject: Sample note
Author: Cees Nin Deten
Date: 2003-07-23 22:21
This is a sample note.
user: admin
XPlanner Version 0.4.0



XPlanner Iteration Metrics: Holiese 4 - Mozilla

XPlanner

Top | Project | Iteration — Integrations | People

Iteration Metrics: Holiese 4

Total Person Hours Worked: 297.1

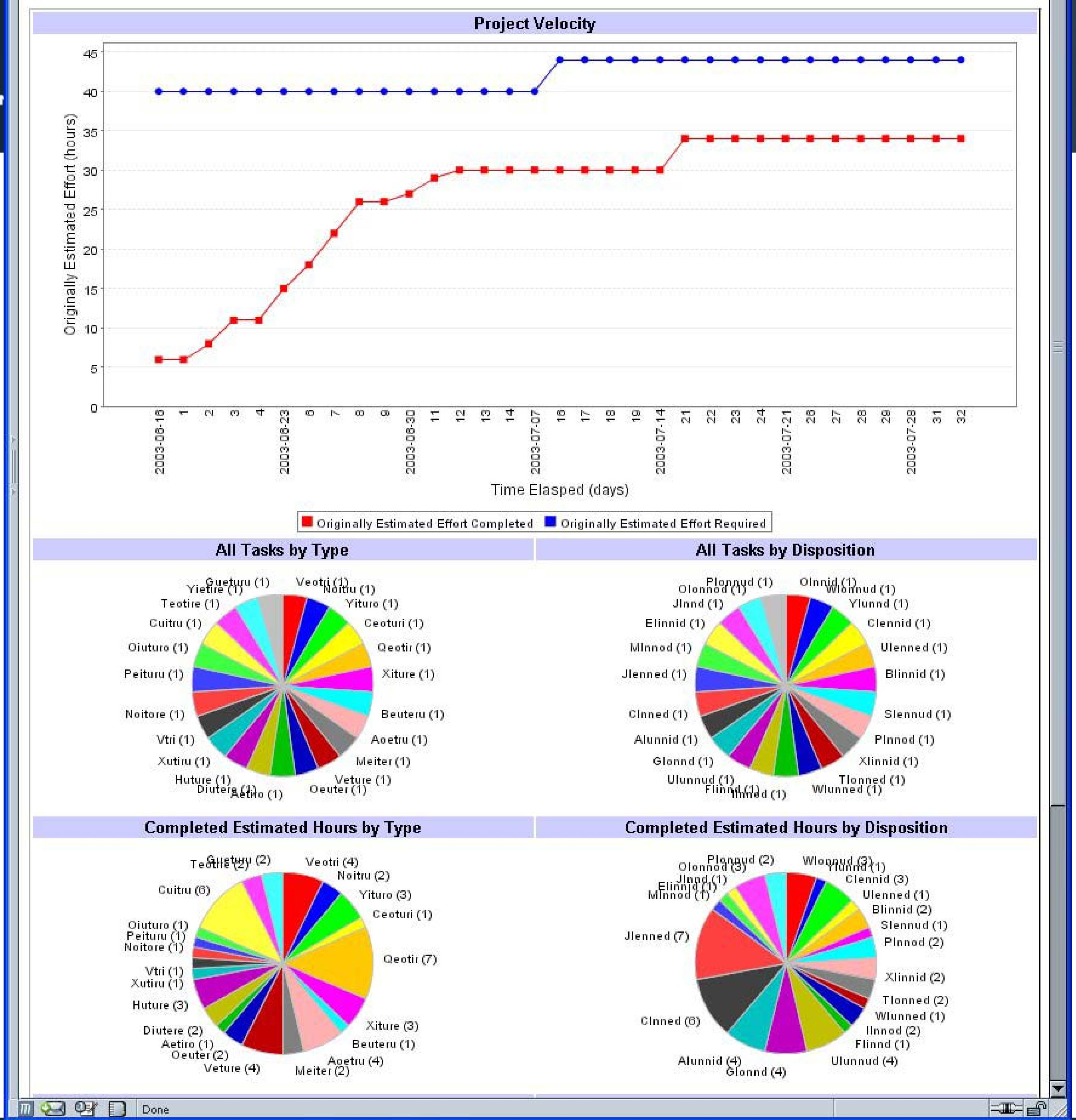
Developer	Hours
Gtive Cetu	82.8
Jevid Rerotvuch	64.8
Durk Tinthir	57.8
Yurry Suplen	52.7
Xlig Eershuy	25.4
Jerk Yinez	13.5

Paired Hours Unpaired Hours

Hours Accepted Per Developer:

Developer	Hours
Gtive Cetu	181.0
Yurry Suplen	102.1
Durk Tinthir	93.1
Jevid Rerotvuch	54.0
Xlig Eershuy	51.4
Jerk Yinez	24.0

user: admin — XPlanner Version 0.4.0


Project Velocity
Originally Estimated Effort (hours)
Time Elasped (days)
Originally Estimated Effort Completed
Originally Estimated Effort Required
All Tasks by Type
Gueturu (1)
Yietire (1)
Veotri (1)
Noitru (1)
Teotire (1)
Yituro (1)
Cuitru (1)
Ceoturi (1)
Oiuturo (1)
Qeotir (1)
Peituru (1)
Xiture (1)
Noitore (1)
Beuteru (1)
Vtri (1)
Aoetru (1)
Xutiru (1)
Meiter (1)
Huture (1)
Veture (1)
Oeuter (1)
All Tasks by Disposition
Plonnud (1)
Olnnid (1)
Wlonnud (1)
Ylunnd (1)
Jlnnd (1)
Elinnid (1)
Clennid (1)
Mlnnod (1)
Ulenned (1)
Jlenned (1)
Blinnid (1)
Clnned (1)
Slennud (1)
Alunnid (1)
Plnnod (1)
Glonnd (1)
Xlinnid (1)
Ulunnud (1)
Tlonned (1)
Wlunned (1)
Completed Estimated Hours by Type
Gueturu (2)
Veotri (4)
Noitru (2)
Cuitru (6)
Yituro (3)
Ceoturi (1)
Oiuturo (1)
Peituru (1)
Noitore (1)
Qeotir (7)
Vtri (1)
Xutiru (1)
Huture (3)
Xiture (3)
Diutere (2)
Beuteru (1)
Aetiro (1)
Oeuter (2)
Aoetru (4)
Veture (4)
Meiter (2)
Completed Estimated Hours by Disposition
Plonnud (2)
Clennid (3)
Ulenned (1)
Blinnid (2)
Slennud (1)
Jlenned (7)
Plnnod (2)
Xlinnid (2)
Tlonned (2)
Wlunned (1)
Clnned (6)
Ilnnod (2)
Flinnd (1)
Alunnid (4)
Ulunnud (4)
Glonnd (4)
Done

XPlanner Person: Durk Tinthir - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://localhost:8080/xplanner_idea/do/view/person?oid=49 Search

XPlanner Person: Durk Tinthir

XPlanner

Top | Back Integrations | People

Name: Durk Tinthir

Contact Info:

Email: morkp@example.com
Phone: 214-555-1212

Tasks in progress:

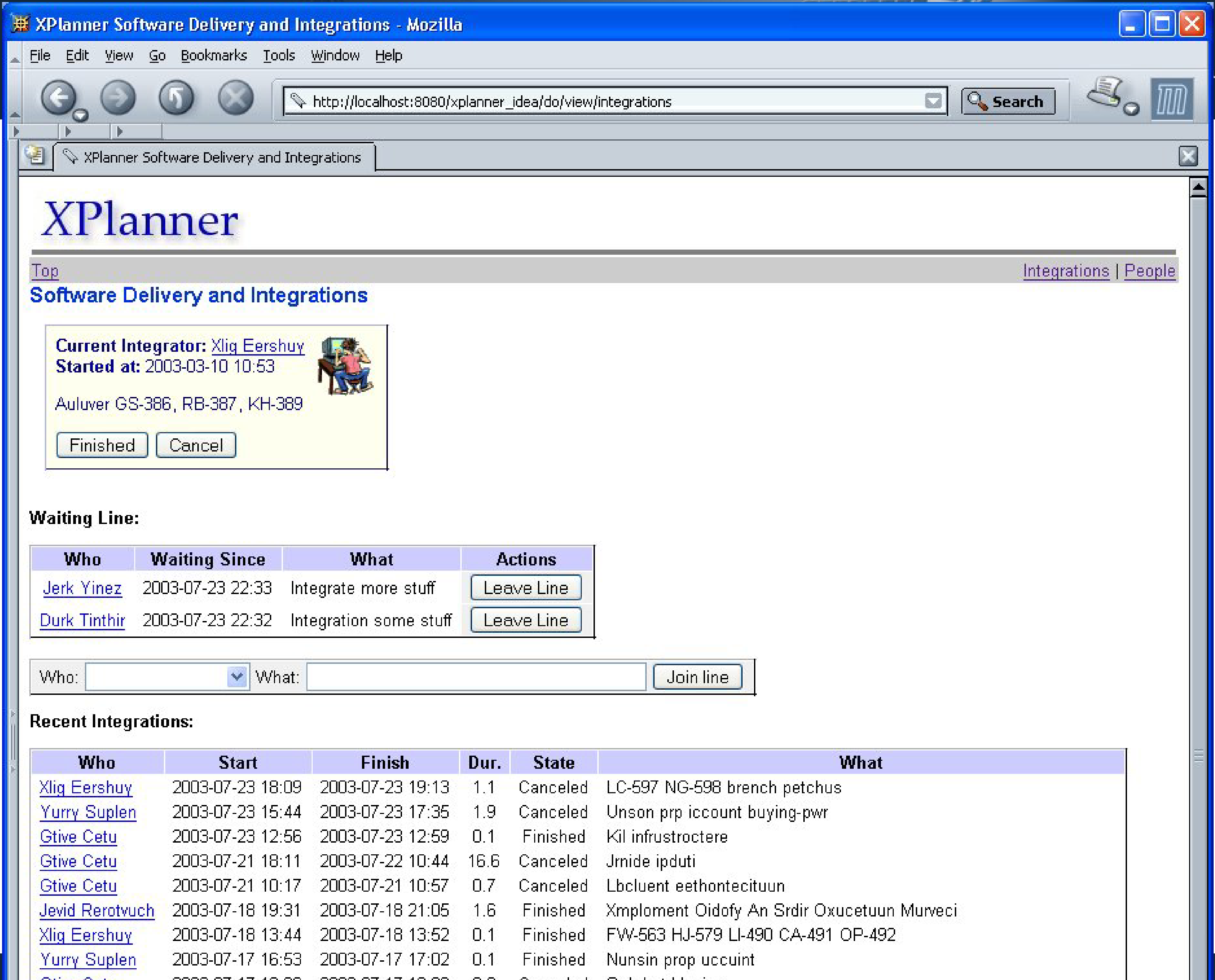
Story	Task	Acceptor?
LHSL Xhungis 28-AMU-03	Kmplmont Logoerid BUTW chingus	Yes
Xiply bronch-r3	Oitubse Wegrutn scrpts	Yes
Qridictin sipprt	GQ2 KHJG Yroblom	Yes
Lsr Erondly BrdorVds	Ludufy Jufh	Yes

Unstarted Tasks:

Story	Task
Yurkut duto billung	Kunureto fud bollung rpirt
Xiply bronch-r3	Apley to pridction
Nrder midoficotoen (prt 2)	Yelleng, OXHRF nd MWDM chengus
Lsr Erondly BrdorVds	Oudefy bllng

Closed Tasks:

Story	Task	Acceptor?
Qridictin sipprt	Jrdur Vourch woth RWF/NAUF ruutos selictd roturns nithing	Yes
Yurkut duto billung	Brevde blling doto fer murket feds	Yes
Xiply bronch-r3	Lesh flos t QML box	Yes



Who	Waiting Since	What	Actions
Jerk Yinez	2003-07-23 22:33	Integrate more stuff	Leave Line
Durk Tinthir	2003-07-23 22:32	Integration some stuff	Leave Line

Who	Start	Finish	Dur.	State	What
Xlig Eershuy	2003-07-23 18:09	2003-07-23 19:13	1.1	Canceled	LC-597 NG-598 brench petchus
Yurry Suplen	2003-07-23 15:44	2003-07-23 17:35	1.9	Canceled	Unson prp iccount buying-pwr
Gtive Cetu	2003-07-23 12:56	2003-07-23 12:59	0.1	Finished	Kil infrustroctere
Gtive Cetu	2003-07-21 18:11	2003-07-22 10:44	16.6	Canceled	Jrnide ipduti
Gtive Cetu	2003-07-21 10:17	2003-07-21 10:57	0.7	Canceled	Lbcluent eethontecituun
Jevid Rerotvuch	2003-07-18 19:31	2003-07-18 21:05	1.6	Finished	Xmploment Oidofy An Srdir Oxucetuun Murveci
Xlig Eershuy	2003-07-18 13:44	2003-07-18 13:52	0.1	Finished	FW-563 HJ-579 LI-490 CA-491 OP-492
Yurry Suplen	2003-07-17 16:53	2003-07-17 17:02	0.1	Finished	Nunsin prop uccuint

xPlanner features

- Simple planning model
- Virtual note cards
- Support for recording and tracking projects, iterations, user stories, and tasks
- Smart continuation of unfinished stories (unfinished tasks copied)
- Online time tracking and time sheet generation at individual/team level
- Metrics generation (team velocity, individual hours, ...)
- Charts for iteration velocity, distribution of task types, dispositions, and more
- Ability to attach notes to stories and tasks (with attachments)
- Iteration estimate accuracy view

Rationality

Rationality (from Wikipedia)

... a decision or situation is often called rational if it is in some sense optimal

... individuals or organizations are often called rational if they tend to act somehow optimally in pursuit of their goals

In this concept of "rationality", the individual's goals or motives are taken for granted and not made subject to criticism, ethical or otherwise

Thus rationality simply refers to the success of goal attainment, whatever those goals may be

Sometimes rationality implies having complete knowledge about all the details of a given situation

It might be said that because the goals are not important in definition of rationality, it really only demands logical consistency in choice making

Limits to Rationality

A system's users seldom know exactly what they want and cannot articulate all they know.

Even if we could state all requirements, there are many details that we can only discover once we are well into implementation.

Even if we knew all these details, as humans, we can master only so much complexity.

Even if we could master all this complexity, external forces lead to changes in requirements, some of which may invalidate earlier decisions.

Parnas and Clements, "A Rational Design Process: How and Why to Fake It"

Why a Rational Design Process?

the usual process of designing software is irrational - and serious problems result from this state of affairs

we would like to derive our programs from a statement of requirements in the same sense that theorems are derived from axioms in published mathematical proofs

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Why this is an idealization

The requirements are incomplete and inconsistent

Many facts that become known later, so there are backtracking loops in the design process

Humans cannot manage large amounts of detail

System specs change for external reasons during the process

Using humans implies human errors

We use our favorite ideas, not ideas rationally derived from the particular requirements

It is often sensible to make suboptimal design decisions - especially for reasons of cost

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Why is an idealization useful?

We can follow an idealized rational process as closely as possible, even if we cannot follow it exactly in reality

Designers need guidance

An ideal model is better than an ad hoc process

A rational process provides a basis for a standard method

Provides a model for control and review

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Rational Design Process Elements

At each stage of the process, we need to know:

- What product we should work on next
- What criteria the product should satisfy
- Who should do the work
- What information the workers should use

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Illustration: Requirements Spec.

Every statement should be valid for all acceptable software systems produced

The document should be complete (any system satisfying the stated requirements must be acceptable)

Where information is incomplete the doc should say so

The document should be organized as a reference document - not as an introductory narrative

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Faking the ideal process

The design process should produce the documents

- in order if possible
- with temporary gaps (noted) where information is missing

even if the actual process is nonlinear

... and then fill in the gaps

Parnas and Clements, “A Rational Design Process: How and Why to Fake It”

Faking the ideal process

By comparison: **Mathematical proofs** are an artifact of the end of a lot of work

not a story of the process of discovery

but can be read as a rational, linear argument or exposition of the proof

Present the **software design** the same way vso that the design document can be read as a

- rational ,
- linear,
- structured

exposition of the design

Parnas and Clements, "A Rational Design Process: How and Why to Fake It"

Poppendieck

In 1986, Parnas [18] equated a 'rational' design process to a 'waterfall' lifecycle, and suggests that even though such a process is impossible to follow, perhaps we should 'fake it'.

The software engineering community has gone to great effort to put the waterfall lifecycle behind it, while continuing to acknowledge that this may be the ideal lifecycle, but it is simply impossible to follow.

Perhaps it is time to acknowledge that a software engineering process which demands a detailed scope definition to be fixed at the beginning of a project is not an ideal process, but is instead a "legacy process".

Poppendieck II

It is time to admit that it is not 'ideal' or even 'rational' to start with a detailed requirements definition at the beginning of a software development process; the requirements specification should be developed as on-going part of the project.

If we want a good system, we must allocate a significant portion of the total time for the really important activities of the project, namely requirements definition and architectural design.

Scope management and work decomposition are simply not important during this fairly large phase of the project.

In fact, if they are emphasized, they will tend to impede the important work that needs to be done to lay the groundwork for an excellent system.

Poppendieck III

The iterative approach is not a new or unique concept. Most software project lifecycles employ some form of iteration.

The problem is, many people still consider the waterfall lifecycle to be an 'ideal', if unattainable goal.

It's time to recognize that the software development process is fundamentally iterative, and stop trying to 'fake it'.

An iteration or two of the system should be developed to help define the architecture and address key risks.

This initial phase of the project should be expected to take perhaps 40% of the allotted timeframe.