



**How to deal with the  
short-blanket syndrome  
while prioritizing the  
roadmap**

*And not feeling so cold*

!y chat

!y mail

!y voice

!y social

!y ticket

!y bot

!y faq

!y monitor

!y reports

!y api

!y command

!y reports

!y api



# Mauricio Navarrete

Graduated from Computer Science at University of **São Paulo**

20 year-experience with Software Development, worked as a Developer, System Analyst. Currently working as a Scrum Master and Agile Coach. Sometimes I help the POs

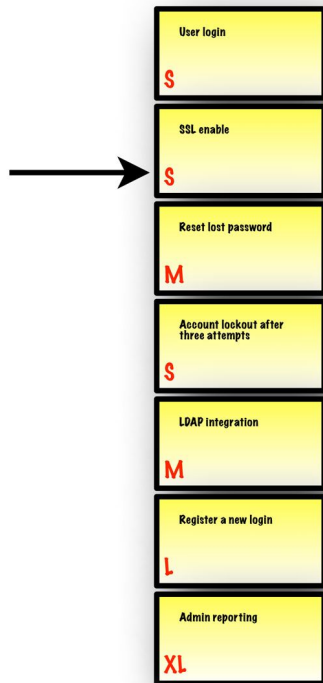
Married

I love travelling





top items  
are more  
granular



only one item  
at a time  
is top priority

# Some context

**Mature company**, not a **startup**

Existing customers and products that need to be constantly maintained

Necessity to develop new products

**Obsolete** technology for part of the **legacy**

Important to **control operational costs** and keep good financial margins



# Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

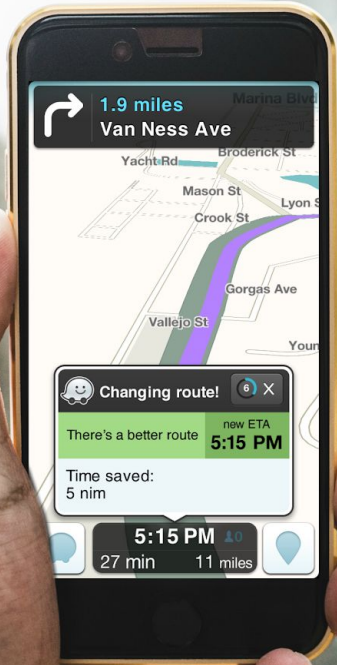
**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

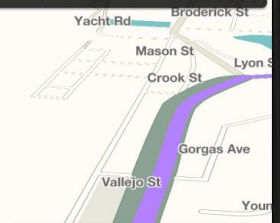
That is, while there is value in the items on the right, we value the items on the left more.

Responding to change over following a plan





1.9 miles  
Van Ness Ave



Changing route!  
There's a better route  
new ETA  
5:15 PM  
Time saved:  
5 min

5:15 PM  
27 min 11 miles



*Cohn, Mike. Agile Estimating and Planning*

DT  E-SERVICE

DT  MAIL

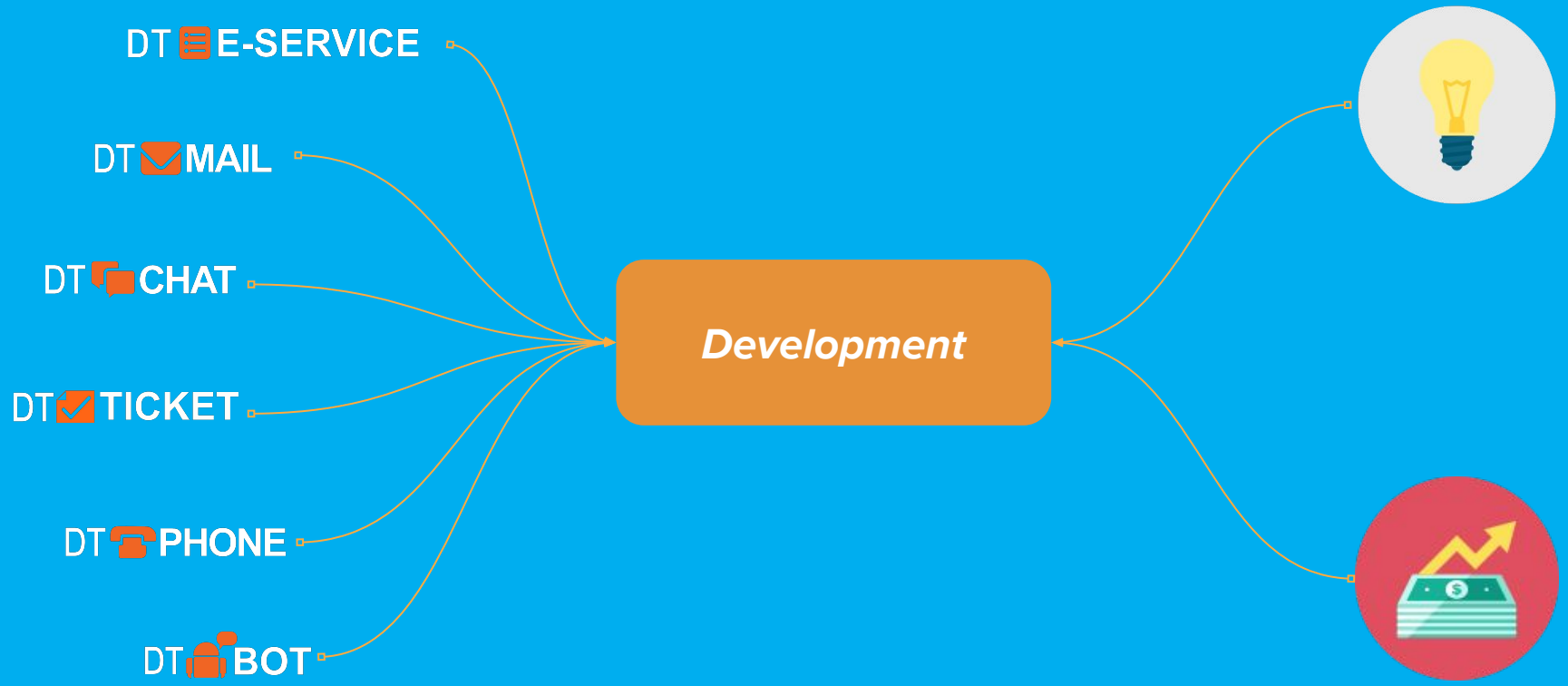
DT  CHAT

DT  TICKET

DT  PHONE

DT  BOT







1st quarter			2nd quarter			3rd quarter			4th quarter		
Front	Percent	Points	Front	Percent	Points	Front	Percent	Points	Front	Percent	Points
Bugs	15%	95	Bugs	15%	95	Bugs	15%	95	Bugs	15%	95
Stability	5%	32	Stability	5%	32	Stability	5%	32	Stability	5%	32
Incremental Evolution	20%	126	Incremental Evolution	30%	189	Incremental Evolution	30%	189	Incremental Evolution	22%	139
Operations: Implementation Process	15%	95	Operations: Implantação e Pós	15%	95	Operations	5%	32	Operations	5%	32
New Products: Integration with Facebook Messenger	15%	95	New Products: DTTicket	30%	189	New Products: DTTicket	15%	95	New Products: BI / Know Base / Template (to be defined)	20%	126
New Products: Intelligent Faq	15%	95									
Deploy	5%	32	Deploy	5%	32	Deploy	15%	95	Deploy	13%	82
Tech Debts: Solving Dependencies	10%	63	Tech Debts	0%	0	Tech Debts: Remove DTMail attachment files from Database	15%	95	Tech Debts: Tenant Microservice	20%	126
<b>Total</b>	<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>

# FIRST STEP

<b>Fronts</b>	<b>Story Points</b>
Bugs	
Stability	
Technical Debt	
Incremental Evolution	
New Products	
Operations	
Deploy	
Innovation	



VDO  
km

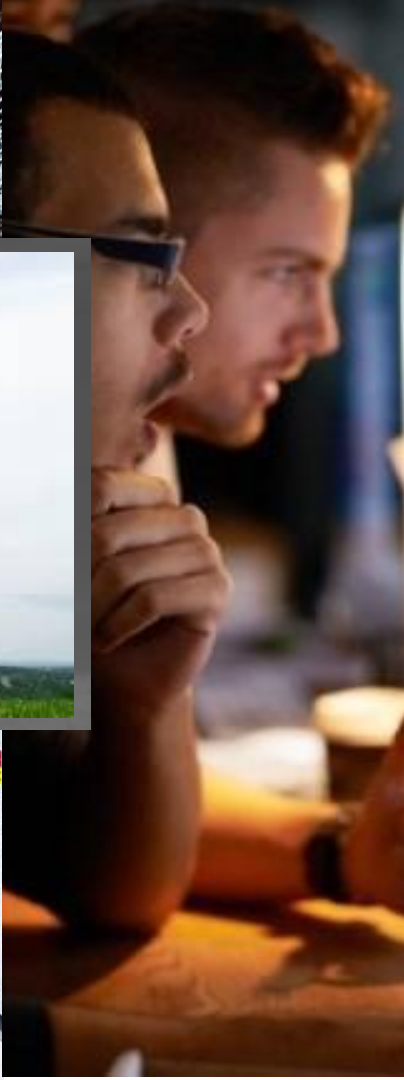
4 1 2 3



P.O.

A person with short, dark hair, seen from the back, is looking at a wall covered in various design sketches, diagrams, and photographs. The person is wearing a light-colored sweater with dark horizontal stripes. The wall behind them is densely packed with papers, some featuring wireframes, flowcharts, and images of people or objects. The overall scene suggests a creative or design workspace.





Company  
**objectives**

# EXAMPLES

- Decrease **churn**
- Increase **product use**
- Improve **NPS**
- **Focus on ecommerce** companies
- Start to **sell** in Latin America

Company  
**objectives**

```
graph LR; A[Company objectives] --> B[Create a backlog for each Front]
```

A flowchart consisting of two blue rounded rectangular boxes connected by a horizontal arrow pointing from left to right. The first box on the left contains the text 'Company objectives'. The second box on the right contains the text 'Create a backlog for each Front'.

Create a  
backlog for  
each **Front**



# Making deliveries more concrete

<b>Backlog Item</b>	<b>Budget</b>	<b>Benefits/Values</b>	<b>Order</b>
List all epics/themes	Some kind of size measure: story points, number of sprints, months, years	More tangible as possible: new sales, money saved. Or benefits like: increases velocity, reduces time lost and so on	If we did one at a time, in which order would we develop?

# Making deliveries more concrete

Backlog Item	Beautiful Name	Cost	Benefits / Values	Cost (Points)	Order
Dependencies among projects / organization of solutions and repositories		1 sprint	Lower development cost, make continuous deploy easier. Decreases bugs related to dependencies	50	1
Data/File	Attachments	3 sprints	<ul style="list-style-type: none"> <li>- Removes DTMail attachments from the database</li> <li>- Decreases infrastructure cost at AWS in U\$ 2.000 / month</li> <li>- Removes part of long requests do transational databases</li> <li>- Pre-condition for History, new GetContacts, E-mail Search</li> <li>- Positive impact on DB performance</li> </ul>	102	2
HealthCheck for endpoints		1 sprint	More agility to diagnise and solve problems	34	3
Tenant	Tenant	3 a 4 sprints	<ul style="list-style-type: none"> <li>- Security</li> <li>- Abstracts Platform concept</li> <li>- Unique URL for system access</li> <li>- Benefits application cluster</li> </ul>	119	4
Service Locator cache, using service locator client		20 points to correct async + 8 for implementing in some projects	Cache for access to service locator. Lowers requisitions to servers, scalability gains, better response time for the main opetarions	28	5

Front	Points
Bugs	X
Stability	Y
Technical Debt	Z
Incremental Evolution	W
New Products	K
Operations	XX
Deploy	YZ
Innovation	WX
<b>Total</b>	<b>7.000</b>

Capacity: **2.400** to **2.800** points a **year**

**2 to 3 years** to develop everything that was **estimated**

# ALTERNATIVES

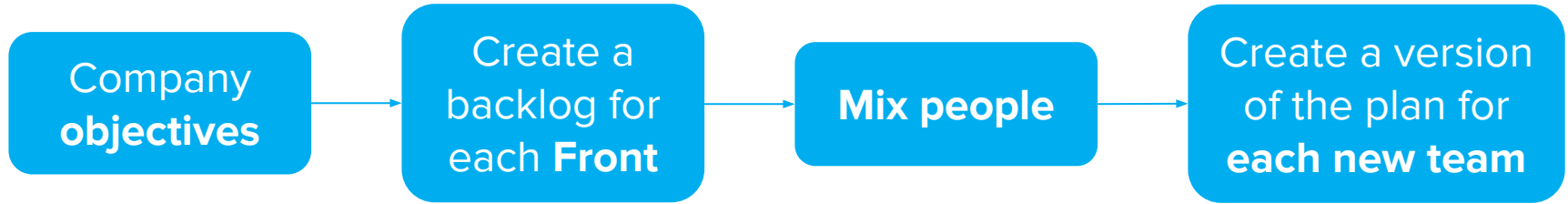
Prioritize, based on **business goals**

Hire **more** people and **grow** your capacity

Check if there are **smarter** ways of developing at a **lower cost** (less history points)

That's why it's so important to **involve more people**









<b>Fronts</b>	<b>History Points</b>	<b>%</b>
Bugs	20	10%
Stability	10	5%
Technical Debt	30	15%
Incremental Evolution	60	30%
New Products	30	15%
Operations	20	10%
Deploy	10	5%
Innovation	20	10%



1st quarter			2nd quarter			3rd quarter			4th quarter		
Front	Percent	Points	Front	Percent	Points	Front	Percent	Points	Front	Percent	Points
Bugs	15%	95	Bugs	15%	95	Bugs	15%	95	Bugs	15%	95
Stability	5%	32	Stability	5%	32	Stability	5%	32	Stability	5%	32
Incremental Evolution	20%	126	Incremental Evolution	30%	189	Incremental Evolution	30%	189	Incremental Evolution	22%	139
Operations: Implementation Process	15%	95	Operations: Implementation	15%	95	Operations	5%	32	Operations	5%	32
New Products: Integration with Facebook Messenger	15%	95	New Products: DTTicket	30%	189	New Products: DTTicket	15%	95	New Products: BI / Knowledge Base / Chat Widget (to be defined)	20%	126
New Products: Intelligent Faq	15%	95									
Deploy	5%	32	Deploy	5%	32	Deploy	15%	95	Deploy	13%	82
Tech Debts: Solving Dependencies	10%	63	Tech Debts	0%	0	Tech Debts: Remove DTMail attachment files from Database	15%	95	Tech Debts: Tenant Microservice	20%	126
<b>Total</b>	<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>		<b>100%</b>	<b>630</b>

# Technical Backlog

Backlog Item	Beautiful Name	Cost	Benefits / Values	Cost (Points)	Order
Dependencies among projects / organization of solutions and repositories		1 sprint	Lower development cost, make continuous deploy easier. Decreases bugs related to dependencies	50	1
Data/File	Attachments	3 sprints	<ul style="list-style-type: none"> <li>- Removes DTMail attachments from the database</li> <li>- Decreases infrastructure cost at AWS in U\$ 2.000 / month</li> <li>- Removes part of long requests do transational databases</li> <li>- Pre-condition for History, new GetContacts, E-mail Search</li> <li>- Positive impact on DB performance</li> </ul>	102	2
Tenant	Tenant	3 a 4 sprints	<ul style="list-style-type: none"> <li>- Security</li> <li>- Abstracts Platform concept</li> <li>- Unique URL for system access</li> <li>- Benefits application cluster</li> </ul>	119	3
HealthCheck for endpoints		1 sprint	More agility to diagnise and solve problems	34	4
Service Locator cache, using service locator client		20 points to correct async + 8 for implementing in some projects	Cache for access to service locator. Lowers requisitions to servers, scalability gains, better response time for the main opetarions	28	5

# Files User Stories

Order	User Story	Cost (Points)
1	Export attachments to S3	5
2	GET /file/{file-id} Operation	3
3	GET /file/{file-id}/content Operation	5
4	DELETE /file/{file-id} Operation	3
5	POST /file Operation	8
6	Implement clients for the operations	5
7	Last adjustments for S3 exporter	5
8	Change email parser to use Files	20
9	Change Admin to use Files	13
10	Change operations used for Agent Module	8
11	Change Specialist to use Files	13
12	Change History to use Files	13
13	Monitoring Production	20

# Important

Tracking the execution

Revision and plan update



# Contacts

[mauricio.navarrete@hiplatform.com](mailto:mauricio.navarrete@hiplatform.com)



[mcnavarrete](#)



[mauricio.navarrete.505](#)



<https://br.linkedin.com/in/mauricio-navarrete-8b11a1/pt>



<https://www.meetup.com/pt-BR/Inovacao-em-Software-como-Servico/> 

<https://medium.com/@dtseekr>



**End**

***thank you!***

***obrigado!***