PWL-BSB # 15:
"Adopting DevOps in the real world: A theory, a model, and a case study" by Luz and colleagues, 2019

Papers We Love Brasília

Presenter: Alessandro Leite

April 28, 2020

DevOps

"It is a set of **practices** and **cultural** values that has emerged in the software development industry."

- ▶ Its claimed benefits include:
  - ▶ better organizational IT performance and productivity

DevOps

"It is a set of **practices** and **cultural** values that has emerged in the software development industry."

- ▶ Its claimed benefits include:
  - ▶ better organizational IT performance and productivity
  - ▶ cost reduction in software life-cycle

DevOps

"It is a set of **practices** and **cultural** values that has emerged in the software development industry."

▶ Its claimed benefits include:
  ▶ better organizational IT performance and productivity
  ▶ cost reduction in software life-cycle
  ▶ improvement in operational efficacy and efficiency

DevOps

"It is a set of **practices** and **cultural** values that has emerged in the software development industry."

▶ Its claimed benefits include:
  ▶ better organizational IT performance and productivity
  ▶ cost reduction in software life-cycle
  ▶ improvement in operational efficacy and efficiency
  ▶ better quality of software products

DevOps

"It is a set of **practices** and **cultural** values that has emerged in the software development industry."

► Its claimed benefits include:
   ► better organizational IT performance and productivity
   ► cost reduction in software life-cycle
   ► improvement in operational efficacy and efficiency
   ► better quality of software products
   ► and greater business alignment between development and operations

▶ It is unclear how to leverage the existing set of tools, practices, and information when trying to adopt DevOps

- It is unclear how to leverage the existing set of tools, practices, and information when trying to adopt DevOps
- Many works have focused on DevOps's characterization instead of providing recommendation practices to assist its adoption

▶ DevOps characterization provides a comprehensive understanding of the elements that constitute DevOps, but do not provide detailed guidance to support newcomers interested in adopting DevOps.

▶ DevOps characterization provides a comprehensive understanding of the elements that constitute DevOps, but do not provide detailed guidance to support newcomers interested in adopting DevOps.

▶ Many practical and timely questions still remain open, including

- ▶ DevOps characterization provides a comprehensive understanding of the elements that constitute DevOps, but do not provide detailed guidance to support newcomers interested in adopting DevOps.
- ▶ Many practical and timely questions still remain open, including
    1. Is there any recommended path to adopt DevOps?

▶ DevOps characterization provides a comprehensive understanding of the elements that constitute DevOps, but do not provide detailed guidance to support newcomers interested in adopting DevOps.

▶ Many practical and timely questions still remain open, including

1. Is there any recommended path to adopt DevOps?
2. Since DevOps is composed by multiple elements do these elements have the same relevance, when adopting DevOps?

▶ DevOps characterization provides a comprehensive understanding of the elements that constitute DevOps, but do not provide detailed guidance to support newcomers interested in adopting DevOps.

▶ Many practical and timely questions still remain open, including

1. Is there any recommended path to adopt DevOps?
2. Since DevOps is composed by multiple elements do these elements have the same relevance, when adopting DevOps?
3. What is the role played by elements such as measurement, information sharing, and automation in a DevOps adoption?

► A DevOps adoption model considering practitioners' point of view

- ▶ A DevOps adoption model considering practitioners' point of view
- ▶ The relevance of the proposed model in a real case scenario

▶ The paper relies on:

► The paper relies on:
  ► Goals / Questions / Metrics (GQM) to both characterizing DevOps adoption (Table 1) and assess the proposed model (Table 2)

Table 1
GQM related to our goal of *characterizing DevOps adoption*.

Analyze	Object under measurement
For the purpose of	*Understand the process of DevOps adoption.*
With respect to	*the objectives that motivate a DevOps adoption process as well as the concerns that might enable DevOps adoption or the concerns that correspond either to the benefits or to challenges related to DevOps adoption.*
From the view point of	*practitioners that have contributed to a previous effort on DevOps adoption.*
In the context of	*companies in different domains that have adopted DevOps.*

Table 2
GQM related to our goal of *assessing the DevOps adoption model*.

Analyze	Object under measurement
For the purpose of	*Understand the relevance of our DevOps adoption model.*
With respect to	*the guidance it provides on the activities that might lift the results of a DevOps adoption effort.*
From the view point of	*practitioners that are participating on a DevOps adoption effort.*
In the context of	*Brazilian Government Institution.*

► The paper relies on:
  ► Goals / Questions / Metrics (GQM) to both characterizing DevOps adoption (Table 1) and assess the proposed model (Table 2)
  ► **Grounded Theory** as the research method to build an explanation about how DevOps has be successfully adopted in different companies

Table 1
GQM related to our goal of *characterizing DevOps adoption*.

Analyze	Object under measurement
For the purpose of	*Understand the process of DevOps adoption.*
With respect to	*the objectives that motivate a DevOps adoption process as well as the concerns that might enable DevOps adoption or the concerns that correspond either to the benefits or to challenges related to DevOps adoption.*
From the view point of	*practitioners that have contributed to a previous effort on DevOps adoption.*
In the context of	*companies in different domains that have adopted DevOps.*

Table 2
GQM related to our goal of *assessing the DevOps adoption model*.

Analyze	Object under measurement
For the purpose of	*Understand the relevance of our DevOps adoption model.*
With respect to	*the guidance it provides on the activities that might lift the results of a DevOps adoption effort.*
From the view point of	*practitioners that are participating on a DevOps adoption effort.*
In the context of	*Brazilian Government Institution.*

▶ The paper relies on:
  ▶ Goals / Questions / Metrics (GQM) to both characterizing DevOps adoption (Table 1) and assess the proposed model (Table 2)
  ▶ **Grounded Theory** as the research method to build an explanation about how DevOps has be successfully adopted in different companies
  ▶ **Focus group** method to explore a real case study when instantiating the proposed method

Table 1

GQM related to our goal of *characterizing DevOps adoption.*

Analyze	Object under measurement
For the purpose of	*Understand the process of DevOps adoption.*
With respect to	*the objectives that motivate a DevOps adoption process as well as the concerns that might enable DevOps adoption or the concerns that correspond either to the benefits or to challenges related to DevOps adoption.*
From the view point of	*practitioners that have contributed to a previous effort on DevOps adoption.*
In the context of	*companies in different domains that have adopted DevOps.*

Table 2

GQM related to our goal of *assessing the DevOps adoption model.*

Analyze	Object under measurement
For the purpose of	*Understand the relevance of our DevOps adoption model.*
With respect to	*the guidance it provides on the activities that might lift the results of a DevOps adoption effort.*
From the view point of	*practitioners that are participating on a DevOps adoption effort.*
In the context of	*Brazilian Government Institution.*

▶ It is a theory-development approach

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:
    - ▶ the absence of clear research hypothesis upfront

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:
    - ▶ the absence of clear research hypothesis upfront
    - ▶ limited exposure to the literature at the beginning of the research

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:
    - ▶ the absence of clear research hypothesis upfront
    - ▶ limited exposure to the literature at the beginning of the research
- ▶ Following a guideline with the following phases:

▶ It is a theory-development approach
▶ Its main characteristics include:
  ▶ the absence of clear research hypothesis upfront
  ▶ limited exposure to the literature at the beginning of the research
▶ Following a guideline with the following phases:
  ▶ **Open coding data collection**: conduct semi-structured interviews with practitioners to search for patterns of incidents concepts and group them into **categories**

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:
  - ▶ the absence of clear research hypothesis upfront
  - ▶ limited exposure to the literature at the beginning of the research
- ▶ Following a guideline with the following phases:
  - ▶ **Open coding data collection**: conduct semi-structured interviews with practitioners to search for patterns of incidents concepts and group them into **categories**
  - ▶ **Selective coding data analysis**: saturate the initial categories and integrate the results into a dense and a consolidated grounded theory

- ▶ It is a theory-development approach
- ▶ Its main characteristics include:
  - ▶ the absence of clear research hypothesis upfront
  - ▶ limited exposure to the literature at the beginning of the research
- ▶ Following a guideline with the following phases:
  - ▶ **Open coding data collection**: conduct semi-structured interviews with practitioners to search for patterns of incidents concepts and group them into **categories**
  - ▶ **Selective coding data analysis**: saturate the initial categories and integrate the results into a dense and a consolidated grounded theory
  - ▶ **Theoretical coding**: explain the categories and the relationships between the categories and reintegrate them into the literature

Table 3

Participant profile. SX means software development experience in years, DX means DevOps experience in years, CN means country of work, and CS means company size (S<100; M<1000; L<5000; XL>5000).

P#	Job Title	SX	DX	CN	Domain	CS
P1	DevOps Developer	9	2	IR	IT	S
P2	DevOps Consultant	9	3	BR	IT	M
P3	DevOps Developer	8	1	IR	IT	S
P4	Computer Technician	10	2	BR	Health	S
P5	Systems Engineer	10	3	SP	Telecom	XL
P6	Developer	3	1	PO	IT	S
P7	Support Analyst	15	2	BR	Telecom	L
P8	DevOps Engineer	20	9	BR	Marketing	M
P9	IT Manager	14	8	BR	IT	M
P10	Network Admin.	15	3	BR	IT	S
P11	DevOps Supervisor	6	4	BR	IT	M
P12	Cloud Engineer	9	3	US	IT	L
P13	Technology Manager	18	6	BR	Food	M
P14	IT Manager	7	2	BR	IT	S
P15	Developer	3	2	BR	IT	S

► direct contact in a DevOpsDays event in Brazil

Table 3

Participant profile. SX means software development experience in years, DX means DevOps experience in years, CN means country of work, and CS means company size (S<100; M<1000; L<5000; XL>5000).

P#	Job Title	SX	DX	CN	Domain	CS
P1	DevOps Developer	9	2	IR	IT	S
P2	DevOps Consultant	9	3	BR	IT	M
P3	DevOps Developer	8	1	IR	IT	S
P4	Computer Technician	10	2	BR	Health	S
P5	Systems Engineer	10	3	SP	Telecom	XL
P6	Developer	3	1	PO	IT	S
P7	Support Analyst	15	2	BR	Telecom	L
P8	DevOps Engineer	20	9	BR	Marketing	M
P9	IT Manager	14	8	BR	IT	M
P10	Network Admin.	15	3	BR	IT	S
P11	DevOps Supervisor	6	4	BR	IT	M
P12	Cloud Engineer	9	3	US	IT	L
P13	Technology Manager	18	6	BR	Food	M
P14	IT Manager	7	2	BR	IT	S
P15	Developer	3	2	BR	IT	S

▶ direct contact in a DevOpsDays event in Brazil
▶ general calls for participation posted on DevOps user groups, social networks, and local communities

▶ What motivated the adoption of DevOps?

▶ What motivated the adoption of DevOps?
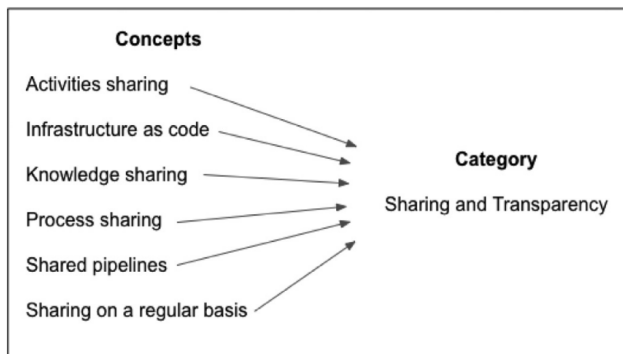▶ What does DevOps adoption mean in the context of your company?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?
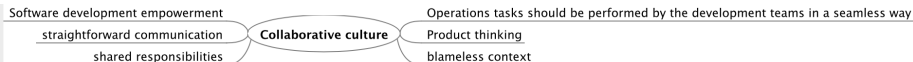- ▶ What were the results of adopting DevOps?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?
- ▶ What were the results of adopting DevOps?
- ▶ What were the main difficulties?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?
- ▶ What were the results of adopting DevOps?
- ▶ What were the main difficulties?
- ▶ What is the relationship between deployment automation and DevOps adoption?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?
- ▶ What were the results of adopting DevOps?
- ▶ What were the main difficulties?
- ▶ What is the relationship between deployment automation and DevOps adoption?
- ▶ Is it possible to adopt DevOps without automation?

- ▶ What motivated the adoption of DevOps?
- ▶ What does DevOps adoption mean in the context of your company?
- ▶ How was DevOps adopted in your company?
- ▶ What were the results of adopting DevOps?
- ▶ What were the main difficulties?
- ▶ What is the relationship between deployment automation and DevOps adoption?
- ▶ Is it possible to adopt DevOps without automation?
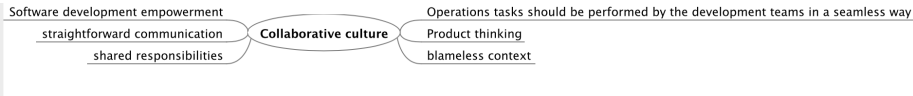- ▶ How has your company fostered a collaborative culture?

Software development empowerment

Operations tasks should be performed by the development teams in a seamless way

straightforward communication

Collaborative culture

Product thinking

shared responsibilities

blameless context

▶ **Why**: Break the silos between development and operations teams

Software development empowerment | Operations tasks should be performed by the development teams in a seamless way

straightforward communication | **Collaborative culture** | Product thinking

shared responsibilities | blameless context
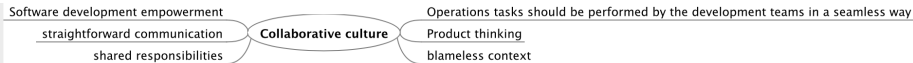
▶ **Why**: Break the silos between development and operations teams

▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day

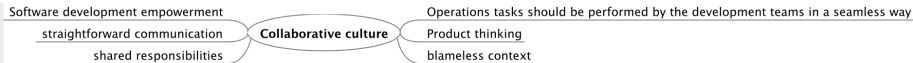| Software development empowerment | | Operations tasks should be performed by the development teams in a seamless way |
| straightforward communication | **Collaborative culture** | Product thinking |
| shared responsibilities | | blameless context |

- ▶ **Why**: Break the silos between development and operations teams
- ▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day
- ▶ **Empowerment**: No need to waiting for the creation of a computing environment or the execution of some scripts

Software development empowerment

straightforward communication **Collaborative culture**

shared responsibilities

Operations tasks should be performed by the development teams in a seamless way
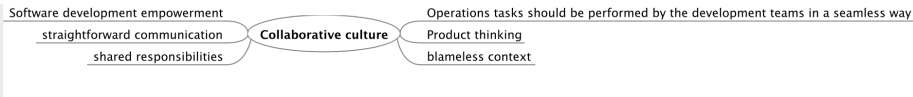
Product thinking

blameless context

▶ **Why**: Break the silos between development and operations teams

▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day

▶ **Empowerment**: No need to waiting for the creation of a computing environment or the execution of some scripts

▶ **Product thinking**: software is a product that does not end after pushing its code to a repository nor when it received for publication

Software development empowerment   Operations tasks should be performed by the development teams in a seamless way

straightforward communication   **Collaborative culture**   Product thinking

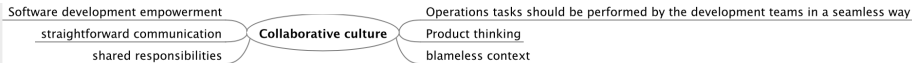shared responsibilities   blameless context

- ▶ **Why**: Break the silos between development and operations teams
- ▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day
- ▶ **Empowerment**: No need to waiting for the creation of a computing environment or the execution of some scripts
- ▶ **Product thinking**: software is a product that does not end after pushing its code to a repository nor when it received for publication
- ▶ **Straightforward communication**: face-to-face communication or through flexible communication channels like Slack or Hip Chat

Software development empowerment

straightforward communication ╱ **Collaborative culture** ╲ Product thinking

shared responsibilities

Operations tasks should be performed by the development teams in a seamless way

blameless context

▶ **Why**: Break the silos between development and operations teams
▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day
▶ **Empowerment**: No need to waiting for the creation of a computing environment or the execution of some scripts
▶ **Product thinking**: software is a product that does not end after pushing its code to a repository nor when it received for publication
▶ **Straightforward communication**: face-to-face communication or through flexible communication channels like Slack or Hip Chat
▶ **Blameless environment**: focus on problem solving instead of witch search; i.e., mistakes are accepted

Software development empowerment | Operations tasks should be performed by the development teams in a seamless way

straightforward communication | **Collaborative culture** | Product thinking

shared responsibilities | blameless context

- ▶ **Why**: Break the silos between development and operations teams
- ▶ **Collaborative culture**: Teams collaborate to perform tasks from the first day
- ▶ **Empowerment**: No need to waiting for the creation of a computing environment or the execution of some scripts
- ▶ **Product thinking**: software is a product that does not end after pushing its code to a repository nor when it received for publication
- ▶ **Straightforward communication**: face-to-face communication or through flexible communication channels like Slack or Hip Chat
- ▶ **Blameless environment**: focus on problem solving instead of witch search; i.e., mistakes are accepted
- ▶ **shared responsibility**: problems do not belong to a unique team

"*In a DevOps adoption, there is a very strong cultural issue that the teams sometimes are not adapted to. Regarding that, one thing that bothers me a lot and that I see very often is people hitching DevOps exclusively by tooling or automation.*" (P9, IT Manager, Brazil)

- **Automation**:

- **Automation**:
  - enables **transparency**

- **Automation**:
  - enables **transparency**
  - ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes

- **Automation**:
  - enables **transparency**
  - ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - increases the **confidence** between teams

- **Automation**:
  - enables **transparency**
  - ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - increases the **confidence** between teams
- **Transparency and sharing**:

- **Automation**:
  - enables **transparency**
  - ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - increases the **confidence** between teams
- **Transparency and sharing**:
  - enables **knowledge sharing** to smooth the learning curve of both technical and cultural necessary to embrace DevOps

- ▶ **Automation**:
  - ▶ enables **transparency**
  - ▶ ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - ▶ increases the **confidence** between teams
- ▶ **Transparency and sharing**:
  - ▶ enables **knowledge sharing** to smooth the learning curve of both technical and cultural necessary to embrace DevOps
  - ▶ allows **activities sharing** through round tables and forum to discuss how tasks are or should be performed

- ▶ **Automation**:
    - ▶ enables **transparency**
    - ▶ ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
    - ▶ increases the **confidence** between teams
- ▶ **Transparency and sharing**:
    - ▶ enables **knowledge sharing** to smooth the learning curve of both technical and cultural necessary to embrace DevOps
    - ▶ allows **activities sharing** through round tables and forum to discuss how tasks are or should be performed
    - ▶ allows **process sharing** through more structured approaches such as lectures and tech talks

- ▶ **Automation**:
  - ▶ enables **transparency**
  - ▶ ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - ▶ increases the **confidence** between teams
- ▶ **Transparency and sharing**:
  - ▶ enables **knowledge sharing** to smooth the learning curve of both technical and cultural necessary to embrace DevOps
  - ▶ allows **activities sharing** through round tables and forum to discuss how tasks are or should be performed
  - ▶ allows **process sharing** through more structured approaches such as lectures and tech talks
  - ▶ creates an appropriate context to have daily meetings with Dev and Ops teams
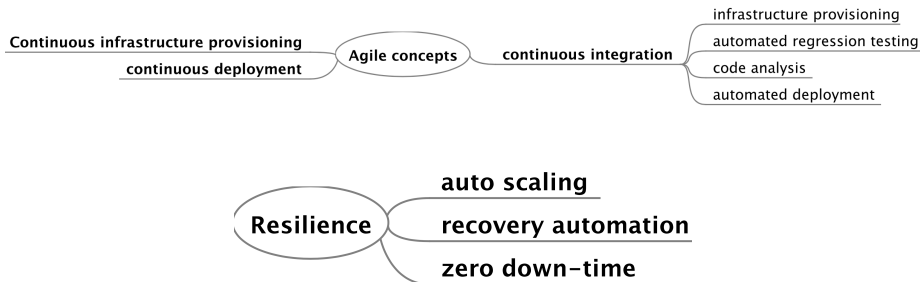
- ▶ **Automation**:
  - ▶ enables **transparency**
  - ▶ ensures **reproducibility** of the tasks, which contributes to reduce rework and the risk of mistakes
  - ▶ increases the **confidence** between teams
- ▶ **Transparency and sharing**:
  - ▶ enables **knowledge sharing** to smooth the learning curve of both technical and cultural necessary to embrace DevOps
  - ▶ allows **activities sharing** through round tables and forum to discuss how tasks are or should be performed
  - ▶ allows **process sharing** through more structured approaches such as lectures and tech talks
  - ▶ creates an appropriate context to have daily meetings with Dev and Ops teams
  - ▶ increases transparency by **shared pipelines**

    > "*The code of how the infrastructure is made is open to developers and the sysadmins need to know some aspects of how the application code is built. The code of our pipelines is accessible to everyone in the company to know how activities are automated*" (P13, Technology Manager, Brazil)

Continuous infrastructure provisioning

continuous deployment

Agile concepts

continuous integration

infrastructure provisioning

automated regression testing

code analysis

automated deployment

Resilience

auto scaling

recovery automation

zero down-time

"When it was necessary to deploy some specific systems, there was often a downtime of a few minutes of the application. In the cases that the deployment did not succeed, the downtime was even greater (perhaps a couple of hours). But with the adoption of DevOps we were able to eliminate the downtime, particularly with the introduction of Kubernetes (https://kubernetes.io/)" (P1, DevOps Developer, Ireland)
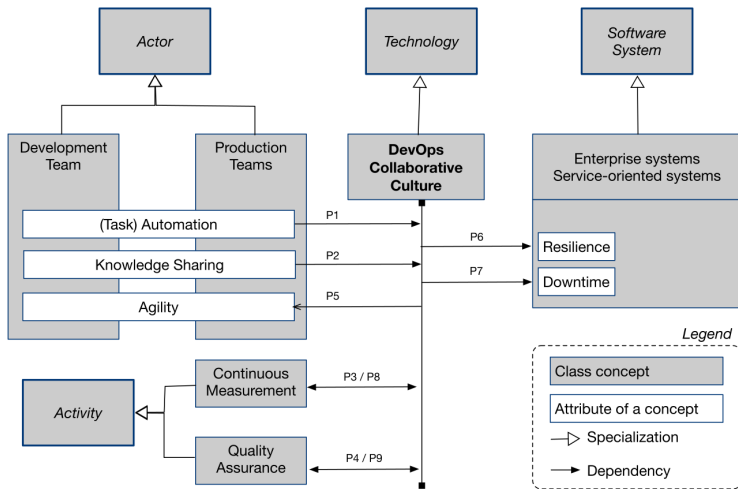
Fig. 2. A theory for DevOps Adoption.

▶ The proposed model comprises three phases:

▶ The proposed model comprises three phases:
  ▶ **knowledge dissemination about the importance of building a collaborative culture** between development and operation teams

► The proposed model comprises three phases:
  ► **knowledge dissemination about the importance of building a collaborative culture** between development and operation teams
  ► **develop the most suitable enablers**. Enablers are the means used to support and develop the collaborative culture

▶ The proposed model comprises three phases:
  ▶ **knowledge dissemination about the importance of building a collaborative culture** between development and operation teams
  ▶ **develop the most suitable enablers**. Enablers are the means used to support and develop the collaborative culture
  ▶ **check the outcomes of the DevOps adoption** to verify the alignment with industrial practices and to explore them according to the company's need
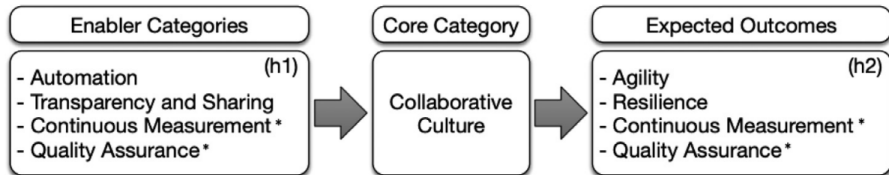
► The engagement and sponsorship of management and the explanations of the process were keys to enable a **collaborative culture**

Table 5
Focus group participants.

P#	Team	Educational background	Experience
P1	Dev	Graduate	3 years in dev team at TCU and 9 years of previous experience
P2	Dev	Posgraduate	6 years in dev team at TCU and 7 years of previous experience
P3	Ops	Graduate	3 years in ops team at TCU and 8 years of previous experience
P4	Ops	Graduate	3 years in ops team at TCU and 10 years of previous experience

Table 6
Focus group topics.

	Topic	Questions
1	Current status of DevOps adoption at TCU	1. What actions developed in the TCU do you consider to be part of DevOps adoption?
		2. What previously existing problems have been solved by these actions?
2	Applicability and utility of the proposed model	1. Do you consider that the proposed model has contributed to DevOps adoption at TCU?
		2. If so, what are the main contributions?
3	Challenges faced and next steps in DevOps adoption	1. What are the main challenges that TCU currently faces in DevOps adoption?
		2. What are the next steps in DevOps adoption at TCU?

▶ Automation of environments' provisions

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers
  - ▶ (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers
  - ▶ (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time
- ▶ Reduction of bureaucracy in the communication between the teams

Assessment of the proposed DevOps adoption model

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers
  - ▶ (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time
- ▶ Reduction of bureaucracy in the communication between the teams
- ▶ Automated database migrations

Assessment of the proposed DevOps adoption model

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers
  - ▶ (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time
- ▶ Reduction of bureaucracy in the communication between the teams
- ▶ Automated database migrations
- ▶ Continuously and automated monitoring of application errors

- ▶ Automation of environments' provisions
- ▶ Use of micro-services and containers
  - ▶ (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time
- ▶ Reduction of bureaucracy in the communication between the teams
- ▶ Automated database migrations
- ▶ Continuously and automated monitoring of application errors
- ▶ Multiple deployment of the applications

Table 4
Deployments of TCU' enterprise systems. MDDS means the maximum number of deployments in a day. MWDS means the maximum number of deployments in a week.

System name	MDDS	MWDS
Autenticidade de Documentos (26 KLOC)	18	37
Cobrana Executiva (33 KLOC)	12	33
Conecta TCU (39 KLOC)	42	51
e-Cautelares (47 KLOC)	9	12
e-TCE (261 KLOC)	29	68
e-TCU Gestores (98 KLOC)	18	64
Mapa de Exposição a Fraude (7 KLOC)	9	39
Ministro (48 KLOC)	17	51
Siga (55 KLOC)	12	17

- Automation of environments' provisions
- Use of micro-services and containers
  - (1) parity between environments, (2) infrastructure provisioning automation, (3) autonomous services, (4) containerization, (5) auto-scaling, (6) recovery automation, and (7) zero down-time
- Reduction of bureaucracy in the communication between the teams
- Automated database migrations
- Continuously and automated monitoring of application errors
- Multiple deployment of the applications

Table 4

Deployments of TCU' enterprise systems. MDDS means the maximum number of deployments in a day. MWDS means the maximum number of deployments in a week.

System name	MDDS	MWDS
Autenticidade de Documentos (26 KLOC)	18	37
Cobrana Executiva (33 KLOC)	12	33
Conecta TCU (39 KLOC)	42	51
e-Cautelares (47 KLOC)	9	12
e-TCE (261 KLOC)	29	68
e-TCU Gestores (98 KLOC)	18	64
Mapa de Exposição a Fraude (7 KLOC)	9	39
Ministro (48 KLOC)	17	51
Siga (55 KLOC)	12	17

- DevOps institutional understanding

▶ DevOps comprises a relationship between **agility**, **automation**, **collaborative culture**, **continuous measurement**, **quality assurance**, **resilience**, **sharing and transparency**

► DevOps comprises a relationship between **agility**, **automation**, **collaborative culture**, **continuous measurement**, **quality assurance**, **resilience**, **sharing and transparency**

► **Collaborative culture** comprises the core element of DevOps adoption

- ▶ DevOps comprises a relationship between **agility**, **automation**, **collaborative culture**, **continuous measurement**, **quality assurance**, **resilience**, **sharing and transparency**
- ▶ **Collaborative culture** comprises the core element of DevOps adoption
- ▶ Automation and tooling are insufficient to enable DevOps adoption in a successfully way

That's all Folks!