International Journal of Engineering Education Vol. 32, No. 5(B), pp. 2260–2273, 2016
Printed in Great Britain

0949-149X/91 $3.00+0.00

An Agile Framework Definition for Creating an Engineering Massive Open Online Course from Scratch: A Case Study*

ANTONIO DE AMESCUA, JOSE MARÍA ALVAREZ-RODRÍGUEZ, MARÍA-ISABEL SANCHEZ-SEGURA and FUENSANTA MEDINA-DOMÍNGUEZ
Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Avd. de la Universidad, 30, 28911, Leganés, Madrid, Spain. E-mail: amescua@inf.uc3m.es, joalvare@inf.uc3m.es, misanche@inf.uc3m.es, fmedina@inf.uc3m.es

Massive Open Online Courses (MOOCs) have emerged as disruptors to higher education bringing the possibility to access learning contents to thousands of students from all over the world. MOOCs are a new way to design and deliver online learning. Learners become part of an on-line community where they can participate as reviewers, collaborate with each other and are engaged in watching videos and other multimedia resources. However, MOOCs are also generating a huge debate around three different aspects: the learning process including evaluation and certification criteria, the lack of skills among instructors to design and plan MOOCs and the technical and security issues of MOOC platforms. More specifically, institutions are currently making a great effort to become part of main facilitators' platforms. They are creating a good number of methodologies, guidelines and best practices to equip instructors with the necessary skills to produce high-quality learning resources that can encourage learners' participation and decrease the dropout rate. On the other hand, engineering education is one of the main areas of interest in MOOC courses. In the software and computer engineering area it is possible to find a huge number of MOOCs in particular topics ranging from an introductory to a master level. However, just a few courses address a holistic view of a domain such as software engineering due to the intrinsic difficulty of summarizing in a few weeks the main concepts of an engineering discipline. That is why, in this paper, authors introduce an Agile MOOC Development Lifecycle (AMDL) to address the challenge of designing a MOOC from scratch. Afterwards, the framework is applied to create a MOOC course about software engineering for a non-technical audience and developed by a large and multidisciplinary team of 18 instructors. This experience is also reported as a case study to validate the proposed development lifecycle. Finally, some discussion, lessons learned and future research lines are also outlined.

Keywords: software engineering; agile; MOOC; on-line education; e-learning

1. Introduction

Recent times have seen the sudden rise of Massive Open Online Courses (MOOCs). The MOOC phenomenon [1] has gained a lot of attraction due to the capability of reaching thousands or even millions of students from all over the world. From facilitators such as the edX, Udacity, Coursera, Khan Academy or XuetangX platforms to course generators, there is a tremendous interest in becoming part of this new learning environment based mainly on interactive multimedia objects such as videos. MOOCs are considered the future of higher and corporate education. Main colleges and universities such as the Stanford University or the Massachusetts Institute of Technology (MIT) as well as large corporations such as Oracle or Microsoft are becoming leaders in the production of this new wave of on-line education through partnerships agreements with the aforementioned facilitators.

However, there is also a lot of discussion and criticism [2] due to the fact that MOOCs could harm higher education since they are primarily driven by financial concerns, not pedagogy. Furthermore, claims are also coming to discuss the quality of MOOCs versus traditional classroom-based education: distance students lose the direct contact with professors, the diverse viewpoints found in classroom and debates are missing and the living experience in the faculty is also lost. Although it seems that from an educational point of view MOOCs are still under study and improvement, it is also clear that MOOCs are also serving to ease the open access to high-quality educational resources to everybody in any part of the world at any time.

On the other hand, and due to the huge amount of MOOCs currently available in the different platforms two new questions arise for course generators: (1) What type of MOOC should be designed? and (2) How the MOOC should be designed and implemented?

The plan, design and implementation of a MOOC are not mere questions of shifting a traditional course to an on-line environment, since most of the problems emerge from two different perspectives: (1) organizational support [3] to deliver a high-impact and quality MOOC. This requires an institutional policy for education technology and meth-

* Accepted 15 May 2016.

odologies for course design apart from the support of the publishing platform and, (2) a well-defined learning process to ease the creation of attractive and pedagogical materials that can mitigate the lack of students participation (engagement [4]), lack of students feedback [5] and the dropout rate [6] and can also fulfill students' needs.

Firstly, most of the institutions engaged in a MOOC platform have clearly defined a set of best practices, guidelines or methodologies [7–10] for the creation of online courses. Thus, it is easy to find in the web with a simple search "MOOC creation guidelines" (Google responds with around 126,000 results, Feb. 2016), a good number of official documents. Apart from blog posts and other resources, universities have made a great effort delivering guidelines (the Northwestern University's [11], the University of Toronto [12] or the University of Edinburgh [8]) easing the task of MOOC design and launch [13]. Moreover, it has been established that one of the cornerstones [8–9] for the successful development of a MOOC lies on the selection of the target audience, the use of multimedia resources, the selection of an adequate assessment method and other criteria that allow platforms to classify the course and, learners, to select the proper course.

Whilst hitherto one of the main efforts have fallen on the creation of guidelines, methodologies and best practices to design and launch MOOCs, from a technological perspective, the second step may rely on defining and planning the learning process within the context of MOOC development. Although the aforementioned documentation can help instructors to address the challenge of delivering a high-quality MOOC, the reality is showing [14] that instructors are facing a real problem to manage the complexity of a complete new paradigm for course design [15]. They need a new mindset to make a paradigm shift from traditional courses to MOOCs, a methodology to plan, design and implement the new contents, skills to produce multimedia resources with new tools, speak in front of a camera and, in general, create a new learning environment. Thus, the development of a MOOC can be actually seen as a project, comprising a team, deadlines, deliverables or activities such as validation and verification, to ensure high-quality contents. Furthermore, and due to the novelty in organizing this type of course, agile principles may help to reach the major objective of fulfilling student's expectations through the delivery of reliable, accurate and valuable contents.

On the other hand, engineering education is indeed one of the main areas of interest in MOOC courses. For instance, in the computer science area, MOOCs focused on specific and vertical topics such as Java, Scala or R as well as others such as mobile programming or data science are the ones having more impact in the on-line community. These domain specific courses are gaining a lot of attraction due to the fact that it is possible to be initiated in a new discipline or to update and broaden your knowledge in others that are being requested in the marketplace. However, just a few courses address a holistic view of a domain because of the intrinsic difficulty of summarizing in just a few weeks the main concepts of a discipline.

Taking into account the aforementioned points, authors address the challenge of designing from scratch a completely new Software Engineering course [48] for a large and non-technical audience. Since software is everywhere, this course is oriented to satisfy the need in the society of understanding and dealing with software-based systems. In particular, the course is a joint effort (team of 18 instructors) of different knowledge areas such as software engineering human computer interaction, interactive design and organization engineering that is supported by the Universidad Carlos III of Madrid (Spain) and the edX platform.

The main contributions of this work are: a) to outline a general framework to implement a MOOC course based on the application of agile principles [16–17], widely spread and used in the software development area [18] as a way to overcome unknown projects and reduce associated risks and, b) to report a proper approach for delivering a high-quality course in a domain overcoming the complexity coming from the following aspects: (1) Materials of traditional on campus courses are not reused. This variable increments the complexity because traditional contents are not designed to be encapsulated while MOOCs contents are commonly knowledge pills and (2) 18 team size: This variable has been introduced in the context of this course development because we believe that to create something that covers a wide knowledge area such as the Software Engineering Profession, it is necessary to count with a variety of professionals with different backgrounds and perspectives of the profession.

2. Related work

The emerging MOOC phenomenon [1] have implied the apparition of a good number of works from different perspectives: (1) learning or pedagogical perspective and classification of MOOCs; (2) design and planning of MOOCs and (3) operational environment: technological, logistical and financial aspects.

In the first case, MOOCs were initially organized into a simplistic classification: CMOOCs (based on principles from connectivist pedagogy; focus on

knowledge creation) or xMOOCs (focus on knowledge duplication). Afterwards, Downes [19] created a four criteria classification: autonomy, diversity, openness and interactivity. Clark [20] also provided a taxonomy of MOOCs based on pedagogy: transfer MOOCs, madeMOOCs, synchMOOCs, asynchMOOCs, adaptiveMOOCs, groupMOOCs, connectivistMOOCS and miniMOOCSs. Colen [2] argued against such classifications to describe the nature of a MOOC and created a set of dimensions to classify and map MOOC courses according to the following criteria: open, massive, use of multimedia, degree of communication, degree of collaboration, learning pathway, quality assurance, amount of reflection, certification, formal learning, autonomy and diversity. She also elaborated the design of MOOC from the 7C's perspective organized in four different layers: (1) Vision: conceptualize; (2) Activities: capture, communicate and collaborate, consider; (3) Synthesis: combine and (4) Implementation: consolidate. The idea behind this classification was to provide a systematic method to design and evaluation quality of MOOCs through different resources and tools that can be applied to each "C". In general, just a very few initiatives discuss the relevant pedagogical aspects of MOOC design [21], how the contents will be delivered to learners [22, 23] and how the interaction between learners will be [24].

Furthermore, the design and creation of MOOCs was also studied to consider learners motivations and goals. For instance, in [25] authors conducted a survey to have a better understanding of students' behaviors. They established that *completers* were more interested in the course content whereas *non-completers* were more interested in MOOCs as a type of learning experience. In the same manner, the need of understanding students' needs was addressed in [26] where authors identified the learning motivations, patterns and factors to influence in student retention and engagement. For instance, they concluded that many MOOC participants do not have the intention of finishing the course (just to access contents) and they also outlined that a MOOC must be a virtual organization comprising education contents, technology and learners rather than a simple combination of materials within a platform. Similar conclusions can be found in other works that try to understand learners to minimize the dropout rate in MOOCs [5–6] by means of data mining techniques.

Apart from the aforementioned best practices, methodologies and on-line guidelines created by main universities and institutions [7–10], the main works in the context of designing and planning of MOOCs can be found in [27] and [28] where authors present a conceptual framework for designing on-line courses and introduce the Business Model Canvas [29] technique to gather the main logistical, technological, pedagogical and financial issues that educators need to think during the design of a MOOC. This framework defines eleven interrelated issues that are addressed through a set of questions, offering a visual and understandable guidance for educators. In [30], authors present another methodology that integrates the learning strategies of xMOOCs and CMOOCs with adaptivity and knowledge management capabilities. The main outcome of this work is a technological framework to design, plan and publish MOOCs. Although all these approaches are completely valid for designing a MOOC from the very beginning addressing the main pedagogical and technological barriers, the development of a MOOC from a project management perspective (coordination, team management, deliverables, stages, etc.) is not completely covered making difficult the monitoring of the development process and the quality assurance of learning contents. Furthermore, the plethora of best practices, methodologies, on-line guidelines techniques, tools and technology related to the design and creation of learning contents make also difficult the selection of the proper development process. Building on these two points, it seems that a project management perspective including agile principles can complete existing MOOC design frameworks [21] easing institutions and instructors to develop MOOCs from scratch.

The present course does not only need to follow existing methodologies but to give an overview of a set of strategic topics for the next wave of software professionals [31]. In this context, existing computer science MOOCs are, in general, focused in specific topics such as "Java Programming" or "Data Science" and those that try to provide a holistic view of an engineering discipline [32] are just presenting existing concepts in a traditional manner, e.g. "Computing: Art, Magic, Science" or "Software Design Principles". That is why, this MOOC also addresses the major objective of reaching a large audience interested in software-based systems in the digital world overcoming the lack of on-line courses in this discipline for non-technical audiences.

3. An agile framework to drive the creation of a MOOC from scratch

As it has been previously introduced, the creation of a MOOC from scratch [27, 28] arises several questions regarding not only the learning process but the plan, delivery and launching of contents. Organizations are making a great effort to create best practices, guidelines and methodologies [7–10]

that try to address the challenge of guiding instructors from the inception of the MOOC until the final delivery (production). The main drawback of all these existing works is that a project management perspective is not completely covered and the implementation of a course can fail due to the lack of management. MOOCs can actually be considered as another type of project and, that is why, it is necessary to apply well-known techniques in the project management area such as the ones highlighted in the agile paradigm [16]. Therefore, a first mapping of the Agile Manifesto Principles [16] to the concept of Agile MOOC has been done to show the commonalities of any agile project and a MOOC, see Table 1.

3.1 AMDL: Agile MOOC development lifecycle

According to Table 1, it seems clear that the Agile Manifesto principles almost fit to MOOC characteristics which also lead us to think on the whole development process of a MOOC as an agile project. Therefore, an Agile System Development Lifecycle (ASDL) [17] can be applied to drive the management of the development process. More

Table 1. Mapping of the Agile Manifesto to MOOC characteristics

Agile Manifesto principle [16]	**Agile MOOC**
"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."	The highest priority of a MOOC also lies on fulfilling students' expectations and needs through valuable learning contents. Continuous delivery is not actually required (just in the development platform).
"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."	Since MOOCs are based on the creation of new and valuable learning contents such as videos, infographics or conceptual maps, it is also required to be ready for changes. Selection of different profiles to validate the generated contents will be necessary to ensure high-quality materials.
"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."	MOOCs are usually planned by weeks. Although it is not necessary continuous delivery what is actually required is to deliver the week contents in a certain slot of time.
"Business people and developers must work together daily throughout the project."	MOOCs also face this situation. The course is likely accepted by an institution (business people) and developers (instructors) are in contact with them to follow the guidelines, ensure quality and meet the final delivery date.
"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."	The creation of a MOOC is an optional activity that institutions offer to their academic staff so only self-motivated people will be actually on board. Moreover, institutions are creating guidelines and offering new training in tools and techniques to produce innovative learning contents.
"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."	Multidisciplinary and large teams are a common practice in MOOC design and planning. This situation implies that face-to-face conversation is actually a very common practice to ease communication and collaboration.
"Working software is the primary measure of progress."	In this case, the main deliverable of a MOOC is a set of videos instead of software.
"Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."	Once a MOOC is launched it is necessary to support the academic team in the engagement of students (e.g. social media strategy), improvement of contents, etc. Furthermore, MOOC contents are expected to persist on time and, depending on the type of course (introductory, intermediate or advance) will also imply the creation of a series of MOOCs.
"Continuous attention to technical excellence and good design enhances agility."	MOOCs are also expected to deliver high-quality contents made by experts.
"Simplicity–the art of maximizing the amount of work not done–is essential."	MOOCs contents are delivered as small pills of knowledge. This implies that a great effort is spent focusing on this task avoiding distractions while designing the learning resources.
"The best architectures, requirements, and designs emerge from self-organizing teams."	MOOCs are usually planned by weeks; the contents for every week can be designed and created by different teams (experts on the week topics).
"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."	The experience designing and delivering MOOC contents must be shared to help others in further developments.

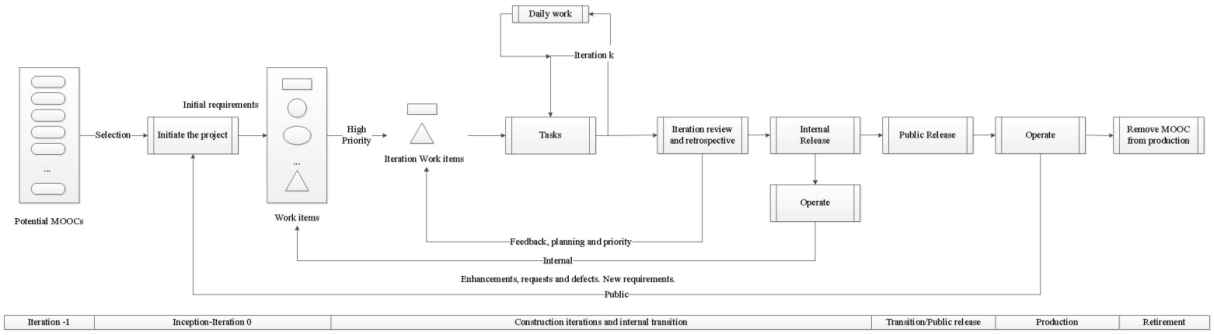

Fig. 1. Agile MOOC Development Lifecycle adapted from [17].

specifically, the different phases of an Agile MOOC Development Lifecycle (Fig. 1), are briefly explained as follows (a new sub process to operate after an internal release has been added and there are now two internal feedback loops):

1. **Concept** (iteration 1-Select the course). This phase tries to identify and prioritize courses in the context of an institution. Thus, it is possible to make a strategy of which areas must be shifted to the MOOC paradigm considering the feasibility and potential impact in the community. It implies that a policy for education technology must be set in the organization in order to evaluate and select the best proposals for MOOCs.
2. **Inception** (iteration 0-Initiate the development of the course). Once a course proposal has been accepted in some area it requires an active participation of stakeholders such as the own organization, professionals in similar areas and potential consumers of the contents (target audience). Here, coordinators of MOOCs must agree the institutional support, build a team and ensure the proper environment for developing contents. They have also to agree a holistic view of the course, to draw learning objectives, to assign team members to the different tasks and to ensure a high-quality course (contents must be complete, correct and consistent) meeting the deadlines of the organization and the MOOC platform.
3. **Construction iterations** (iteration i..k-Deliver work products meeting the changing needs of stakeholders). To do so, coordinators must engage stakeholders in the verification and validation of course contents (texts, videos, questionnaires and any other multimedia resource). Furthermore, they have to create a collaborative environment to boost cooperation between team members, to share experiences, to make tests of the generated contents and to make documentation of the whole development process. The main outputs of these iterations are the different versions of learning resources generated by instructors. Iterations will release these contents to an internal MOOC platform similar to the final target platform.
4. **Transition** (Release). This stage represents "*the end of the game*". All contents are available in the internal platform and can now be deployed in the target facilitators' platform. Once the course is available, last tests will be performed to ensure that all contents are correctly deployed in the production environment. In general, MOOC contents will be partially released after finishing iterations and, at least, two complete releases will be done before starting the course: (1) full internal release and (2) full release in the target platform.
5. **Production** (Operate and support). The course will be launched after an engagement campaign and it will be in production until finishing all the planned weeks. Here, the facilitators, the organization and the team must ensure the proper execution of the course measuring the impact of contents, encouraging learners' participation and becoming active members of the community created around the course. This phase actually means "*the start of the game*" and it requires the commitment of all actors involved in the development of the course.
6. **Retirement** (Remove the system completely). This stage is still not very clear in the context of MOOCs. A MOOC is a kind of product that will persist on time and the complete retirement will only occur in case of not getting attraction, low-quality contents and/or low participation and engagement of learners. Taking into account that these situations will be mitigated from the iteration -1, it implies that a MOOC will not be retired in a short or mid-term. However, new versions and enhancements of the MOOC will be released as well as *spin-offs* of specific topics.

Table 2. Mapping of Agile Project Development Tasks to Agile MOOC actions

Stage and Goal(s)	Agile Project Development Task [17]	Agile MOOC action
Concept Define a strategy and policies at institutional level	*Define the business opportunity*	• Need of delivering new learning resources for new concepts or reinforcing existing topics • Improve the corporate image • Dissemination and spread of knowledge in key areas for the institution • Cost and effort estimation
	Identify a viable for the project	• Select between reuse existing (classroom-based) courses or create a new one from scratch • Select the strategy to build the course (people, materials and equipment): own resources, partnership with other institutions or buy existing resources
	Assess the feasibility	• Feasibility analysis: economic feasibility, technical feasibility, operational feasibility, and political feasibility • Study of potential risks and mitigating factors
Inception Initiate the MOOC project	*Garnering initial support and funding for the project*	• Realistic feasibility analysis including cost and effort estimation for a particular MOOC [33]
	Actively working with stakeholders to initially model the scope of the system	• Define target audience and topic • Define high-level needs for creating this new MOOC • Understand the main goal of the MOOC (not generate documentation since specific requirements will be gathered in a JIT (Just-In-Time) manner through brainstorming, cards and other collaborative techniques)
	Starting to build the team	• Identify required skills of your team members • Identify key members: multidisciplinary teams • Identify key roles [34]: coordination, design of contents, quality controllers, on-line instructor, social media strategists, etc.
	Modeling an initial architecture for the system	• Make a draft of the course syllabus
	Setting up the environment	• Equipment: recording rooms, television studio, camera, audio and video materials, software licenses, etc. • Train your team on the new communication channels (e.g. camera)
	Estimating the project	• Building on the previous items, make a first estimation in terms of time and costs
Iteration k Deliver high-quality contents	*Collaborating closely with both our stakeholders and with other developers*	• Define your stakeholders for this course: team members, existing students, institution representatives, technicians, relatives, friends and colleagues • Reduce risks by getting the feedback of potential learners such as your stakeholders • Keep a close collaboration through a defined communication channel
	Implementing functionality in priority order	• Trust on the opinion of your stakeholders to prioritize the contents to be generated
	Analyzing and designing	• Analyze the particular requirements to create a learning resource by model storming on JIT basis for a few minutes before spending hours preparing a learning resource • Establish quality criteria to verify and validate both: contents and their delivery
	Ensuring quality	• Follow institutional guidelines (if any) • Refactor your contents (change the script, videos, images or any other multimedia resource used to present some concept)
	Regularly delivering working solutions	• Try to generate a complete lecture as soon as possible. The sooner, and more often, you can do it the better (testing and quality). • Publish your lecture to the internal platform
	Testing, testing, and yes, testing	• Do the thing right (personal verification of your lecture) • Do the right thing (external validation of your lecture by stakeholders)
Transition Deliver the solution into production	*Final testing of the system*	• Complete validation of a week within a platform • Complete validation of the course within a platform
	Rework	• Find and fix issues and errors
	Finalization of any system and user documentation	• Generate a document containing the history of the development process • Generate a document that can serve as a learning resource containing the lectures, images, etc. any resource that is not video. Printable version of the course.
	Training	• Support to the final users of your MOOC (learners) through guides and examples to accomplish with the tasks • Support to the technicians that will operate your course in production
	Deploy the system	• Prepare a three level environment for deploying the course: development, integration and production • Create a checklist to automate the task of checking that any resource, questionnaire, reading resource and, in general, any link or resource is accessible
Production Keep the system useful and productive for end users	*Deploy the system into production*	• Deploy the MOOC course in the facilitators' platform • Execute the aforementioned checklist to ensure that everything is ready • Join your own course • Define and put in action your social media strategy • Get the engagement and participation of learners
Retirement (Partial or full) Removal of the system in production	*Strategy and agreement for MOOC maintenance*	• Select the proper strategy to update, replace or extend your MOOC: – The MOOC will be completely replaced. – The release is no longer supported. – The MOOC no longer meets institution strategy. – The MOOC is redundant. – The MOOC has become obsolete.

Although the different phases of an AMDL looks very much to a traditional system development lifecycle, the application of this lifecycle results in the creation of a highly collaborative, iterative and incremental development environment. Team members will work closely with stakeholders to understand their needs, implement the required contents and test the final outcomes to get early and quick feedback. They will be equipped with full life cycle skills to play any role in the AMDL such as technical coordinators, designers of contents or quality control.

More specifically, each stage in a common ASDL can be mapped into an AMDL action or task that will be required to ensure that the life cycle is properly applied to the development of a new MOOC. Table 2 presents the aforementioned stages including an overall goal for each one. It also includes a set of tasks following the methodology presented in [17] to finally propose a set of actions that turns the development of a MOOC into Agile MOOC practices. While the two first stages will be managed by the institution and course coordinators, iterations will generate the proper MOOC contents that will be released in a development environment (a MOOC platform). Thus, any resource will be always available for verification and validation until its final release in the production environment (the MOOC facilitators' platform). The AMDL approach eases instructors and, more specifically, course coordinators, to tackle the development of a MOOC from an agile project management perspective in which each stage has a clear list of practices (not exhaustive) that will generate the different MOOC assets and documentation.

4. Case study: a Software Architect course

Software engineering courses [35, 36] have been traditionally designed for people with some background or interest in programming and other computer science related areas. However, this situation has dramatically changed. Software is contributing to create smart environments such as cars, cities or transportation systems. In the coming 20 years, everything will be interconnected and software will play an important role not only in sustainable economic development but also in human development easing our daily life and improving the welfare state. Thus, software and technological breakthroughs will be the main drivers [37] to the knowledge and digital society. Besides, software is not anymore a computer program, there is much more at stake than a simply set of programming instructions that is knowledge. Software is becoming a commodity, an intellectual and organizational asset that is embedded in products, business, manufacturing processes, etc. and serves to transform a simple organization in a leaning organization. That is why, all agents involved in software thinking or governance of software-based systems will require new and particular skills (digital eSkills) that differ from other engineering disciplines. Obviously, new educational and learning methods/techniques [38] will be also necessary to equip this new wave of professionals with the required *eSkills*.

4.1 Application of the ADML to create the ''The Software Architect Code: Building the Digital World'' MOOC

The previous section has outlined a general framework for the development of MOOCs applying agile principles. In this case study, these techniques are applied to create the MOOC entitled as ''The Software Architect Code: Building the Digital World'' which main aim is to enhance the software capabilities of any individual or organization interested in a software-defined world through an introductory course.

Table 4 depicts a summary of the outcomes related to the agile tasks. In order to set the ''Concept'' of the project, it is necessary to emphasize the effort carried out by the Universidad Carlos III de Madrid (UC3M), Spain. The institution has recently established a strategy for on-line education implemented through different policies to boost education and technology. The main outcome of these policies, in the context of MOOCs, is a yearly call for course proposals. The present course was submitted as a complete new course and accepted in the first call (Nov. 2014) with the aim of presenting a holistic view of the software world and bringing the building blocks and notions regarding software to a large and non-technical audience.

To do so, the course was designed to fulfill the following learners' needs:

- Gain the skills needed to succeed as a software designer and software architect.
- Understand the role of a software architect in the digital world.
- Learn the importance of capturing all essential user needs and applying the ''Do it yourself'' method to plan and build software pieces.
- Explore the software design and testing activities addressing the new challenges of tech-focused times
- Understand new interaction paradigms, creativity and design methods that move the focus from the software to the human who use it.

In order to convey these skills, an executive board comprising Full Professors in different areas such as software engineering, requirements and knowledge

engineering, software design and testing, human and computer interaction designed the course syllabus dividing the eight weeks of the course into four different parts:

- PART I: Decoding the talents of an architect of the digital world: "*Go to the renaissance to discover the Da Vinci principles and come back to the future to unveil the talents required for building a digital world.*"
- PART 2: Envisioning needs and wishes: "*The art of understanding different prospects to classify and prioritize needs and wishes.*"
- PART 3: Discovering how to build and test an idea: *"Drawing sketches and polishing their design. Looking for perfection.*"
- PART 4: Shifting the focus from software-centric to human-centric development: "*Get to know the main concepts and principles of human computer interaction, creativity and participatory methods that a modern software engineer should be familiar with.*"

In order to ease the communication between instructors and the administrative staff, the University established the use of Google Drive and Google Docs as mandatory to share any item generated during the development of the course. Thus, the administrative staff, technicians and any other stakeholder could access contents keeping a track of any change and boosting communication and collaboration. A key point for the proper development of the course was to initially equip team members with professional training to speak in front of a camera and to use tools for editing video. Team members could then work independently and make their own tests before recording and releasing a video lecture.

Furthermore, it was established a two round process for reviewing contents: (1) verification, to ensure that a particular lecture was perfectly aligned to objectives and topics of that part and (2) validation, to ensure that such part was also aligned to the overall objective and topics of the course. In general, the time spent to release a video lecture and the related contents averaged between 15–20 hours implying a total number of 1120 hours of effort to deliver 56 video lectures and related contents such as questionnaires and infographics.

The course is currently in the production stage (open and publicly available within the edX platform). As an example of the generated contents Fig. 2 shows a part of a video including a visual and interactive map and Fig. 3 depicts the infographics associated to that video. However, it is also important to highlight that the publication of the course is not the "*end of the game*" but the start. That is why, once contents are ready the next step consists on establishing a dissemination and engagement strategy to get the attraction of potential learners and boost participation during the execution of the course. To do so, the official edX newsletter and social networks (e.g. a Twitter account: @swarchitectedx) have been the two main channels to promote and create a community around the course. According to the metrics provided by the "Insights edX tool", the course has a current enrollment of 7810 students, see Fig. 4, coming from 154 different countries being United States (18.3%), India (12.9%) and, United Kingdom (3.9%) the top three source countries of students. From a demographics perspective, see Table 3, the median student age is 29, 73.8 % of them have a college or higher degree and 84.1% have reported to be male. These data imply that learners are already professionals (or have some experience) and they are coming mainly from well-stablished technological countries (United States and United Kingdom) and emerging ones (India). However, the percentage of

Fig. 2. Screenshot of the on-line video" Week 1: The Da Vinci Code": *Lecture 1.2: Sensazione talent.*

Fig. 3. Related Infographics of the on-line video" Week 1: The Da Vinci Code": *Lecture 1.2: Sensazione talent*.

female students seems to be very low taking into account that is actually a general-purpose course for gaining digital skills. Finally, just 16 students have already verified their enrollment although it is expected to have an increment of this value before finishing the course.

4.2 Discussion and lessons learned

The development of a MOOC from scratch has led to face different barriers and challenges from both perspectives: learning process and project management. Although the institutional support and the strong background of all instructors, the creation of a complete new MOOC in a wide area such as software engineering, has implied the need of a new mindset to address the challenge of summarizing in just a few weeks and, for a non-technical audience, an engineering discipline. More specifically, the lessons learned during the development of this MOOC can be summarized from different perspectives as follows:

- From an engineering education perspective, the software engineering area is becoming a crucial discipline that everybody needs to know. Everything is fueled by software and "*Software is eating the world*" as the popular web pioneer Mark Andreessen pointed out in an article in the New York Times in 2011. We have seen that cars are now smart cars, everything is connected to Internet to communicate and collaborate each other. Large companies in different sectors are becoming software companies. Traditional activities are turning into software-defined tasks and, we, ourselves, have some software inside through wearables. Therefore, software is actually eating the world. That is why; it is also possible to state that we are living an "industrial revolution", the Industrial Revolution 4.0. All indicators suggest that we are going to dive into the age of the smart environments where objects communicate each other for our own benefit, improving our workplace, home, city and daily life activities.

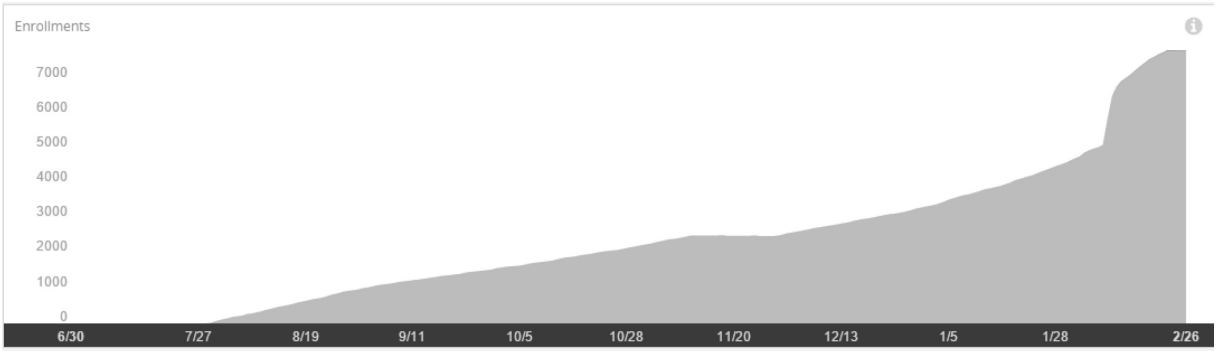

Fig. 4. Daily student enrollment between July 2015 and February 2016.

Table 3 Demographics (age, education and gender) of the "Software Architect" MOOC course

Metric/Topic	Age	Metric/Topic	Education	Metric/Topic	Gender
Students 25 and Under	30.6%	High School Diploma or Less	23.3%	Male	84.1%
Students 26 to 40	53.7%	College Degree	43.6%	Female	15.3%
Students 41 and Over	15.7%	Advanced Degree	30.2%	Not reported	0.5%

Table 4. Summary of the actions taken to develop a MOOC course applying agile principles.

Stage	Agile Project Development Task [17]	Agile MOOC outcome
Concept	*Define the business opportunity*	• Creation of a governing body for strategy and digital education at University Carlos III de Madrid • Creation of the UTEID (Educational Technology and Innovative Teaching Unit) in UC3M • Institutional document [43] including the policies and the proper budget to promote the creation of MOOCs. E.g. UC3M) Digital • Yearly Call for MOOCs proposals[44] requirements, vision and mission of the course
	Identify a viable for the project	• Proposal of a complete new MOOC entitled: "The Software Architect Code: Building the Digital World" • Joint effort of a multidisciplinary team of 18 members belonging to different knowledge areas in UC3M
	Assess the feasibility	Document assessing the need of the new course including a feasibility analysis
Inception	*Garnering initial support and funding for the project*	• Course syllabus document and initial plan: – 4 parts, 8 weeks, 56 video lectures (between 6–8 minutes) – Bi-weekly sprint
	Actively working with stakeholders to initially model the scope of the system	• Target audience: future software engineers and any person interested in software • Type of MOOC: introductory *madeMOOCs* • Main objective: equip with the required software skills to any professional *"Let's speak software"*
	Starting to build the team	• Required skills: software, requirements and knowledge engineering, software design and testing, human and computer interaction • Members: a multidisciplinary team of 18 instructors • Roles: MOOC coordinators, part coordinator, teacher, design of contents, quality controllers, on-line instructor and social media strategists
	Modeling an initial architecture for the system	• Draft of a course syllabus
	Setting up the environment	• Equipment: recording rooms (2), television studio (1), software licenses for Camtasia Studio, Go Animate and Explee. • Training courses: 1) "Lessons to speak in front of a camera" (20h) and 2) "Video editing with Camtasia Studio" (10 h)
	Estimating the project	• Duration: March 2015 to December 2015 (9 months) • Time estimation: around 56*20 hours per video lecture = 1120 hours • Total cost estimation (development): between 35.000–40.000 € (similar to results in [14])
Iteration k	*Collaborating closely with both our stakeholders and with other developers*	• List of stakeholders: 18 team members, around 36 relatives and friends, 4 UC3M administrative staff members and 6 UC3M technicians • Collaboration through Google Drive (institutional tool).
	Implementing functionality in priority order	• Validation of contents in the context of a week, part and course coordinator.
	Analyzing and designing	• Daily meeting (5–10 minutes) to review the current status (week level) • Bi-weekly meeting (up to 30 minutes) (part level) • Monthly meeting (1 hour) (course level)
	Ensuring quality	• Following the UC3M guidelines for developing a MOOC: resources, licenses, software, etc.
	Regularly delivering working solutions	• Bi-weekly sprint to generate 2 videos and related resources • Publication of results in the UC3M internal MOOC platform
	Testing, testing, and yes, testing	• Part members verification • Course members and stakeholders validation
Transition	*Final testing of the system*	• Checklist to validate week contents and the full course
	Rework	• Not applicable
	Finalization of any system and user documentation	• Printable document of course contents
	Training	• Not applicable
	Deploy the system	• UC3M internal MOOC platform and edX • Checklist to ensure everything is working properly
Production	*Deploy the system into production*	• Deployment on December 2015 • Starting date: February 2016
Retirement	Strategy and agreement for MOOC maintenance	• Not applicable

In this context, it is necessary to equip people with the required skills to manage the main notions, vocabulary and activities related to software engineering. However, existing software engineering courses are somehow isolated, specialization of software-related topics have implied the compartmentalization of knowledge and only specialists in such topics are able to understand all the notions behind software and to have a holistic view of the software development process. Since software is becoming a reality, and a commodity, in any sector and due to the lack of software courses for a non-technical audience, this course represents a big step towards: (1) the socialization of the software engineering discipline; (2) the dissemination and spread of software concepts to everybody and (3) the training of a new wave of professionals.

To do so, a great effort has been done to design a complete new course that compiles the cornerstones and building blocks of the main software engineering areas [39] in just one stop. A multidisciplinary team of 18 instructors has been formed to effectively design, devise and deliver pills of knowledge related to software engineering concepts. Furthermore and taking into account the new on-line learning environment, the course has been packaged as an introductory massive and open on-line course [40] to spread software concepts over the world. In conclusion, we have addressed the challenge of spreading the knowledge about software to a large and non-technical audience taking advantage of the new possibilities provided by MOOCs [41]. ''The Software Architect Code: Building the Digital World'' represents one of the first on-line courses to enable people to ''*speak software*''.

• From a learning process perspective, we have seen that the creation of a MOOC from scratch requires a great effort [14] to plan, design and create new contents. In traditional classroom based education, the teacher is leading the learning process through face to face lectures in which students can interact with each other and discuss with the teacher at any moment. This really enriches the learning process and students can easily acquire concepts that are reinforced through homework, team work, presentations and other activities [38].

However, this situation completely changes in an on-line learning environment. MOOCs must be designed to ensure that learners can acquire the same concepts without the direct communication between them and the teacher. Although MOOCs platforms provides the proper mechanisms to boost the engagement and participation of learners in forums and other communication channels, a MOOC, as a course, gives learners an off-line and isolated learning experience. Learners are just watching a video or other multimedia resource that must convey the main notions of a topic and sparkling new ideas. This implies that course contents must be carefully designed and created [3].

To do so, quality is a must factor in a MOOC and instructors really need to acquire new skills to fit to this new environment in which they will delegate their leading to a multimedia resource that can be *closed* at any time increasing the dropout rate and other common issues [42] in MOOCs. In order to mitigate the dropout rate [5], high-quality resources must be provided and instructors need to learn how to speak in front of camera, design new, attractive and graphical contents and create a good performance to present any relevant resource.

Regarding the time and effort to produce the new contents, we have also found that the completion of an 8–10 minute video lecture and related materials can imply up to 20 hours compared to the averaged 8 hours that would take to prepare 1 hour of a complete new traditional classroom lecture. This means that under the same circumstances, the preparation of 1 hour of on-line video and related contents implies up to 15 times (120/8) more effort in terms of time than the preparation of 1 hour in a traditional lecture.

Furthermore, and during the development of this MOOC, two new questions were raised: (1) Who should play the role of instructor in a MOOC? A professional actor or an instructor? and (2) Is a MOOC a new kind of theatrical genre?

Both questions have generated a huge internal debate but we finally agreed that although a professional actor could make a better performance, the instructor or expert in some topic will convey ideas in a better and motivated way. However, it was also agreed the need of equipping instructors with actor skills to overcome the barrier of speaking in front of a camera. On the other hand, MOOCs pretend to become a revolution in terms of on-line learning so it is not recommendable to keep the same habits that can be found in the traditional classroom lectures. Innovation is completely required and this also implies that techniques and materials to teach should be changed, improved or re-designed. It is not a matter of conveying cold facts but learning experiences.

• From a project management perspective, a general framework to apply agile principles for developing MOOCs has been outlined. A MOOC can be seen as a project in which it is necessary to meet deadlines, manage different types of resources and produce a set artifact.

Moreover, the present MOOC is slightly different to other courses in which a small team ranging from 2 to 4 members collaborates to produce the materials for a specific topic. In this case, we have coordinated a team of 18 members. We have also

bet for presenting an overview of an engineering discipline such as software engineering. To do so, the course has been organized in 8 weeks allocated in 4 different parts generating a total number of 56 videos and other related resources. Taking into account these characteristics and the time frame to develop the MOOC, we have made a strong effort and we have seen the necessity of applying agile principles in the development and management of this MOOC. As main outcome of this first experience, we have acquired a strong background in the art of developing of MOOCs [10] that is translated into a set of agile principles that can ease instructors to plan and develop future MOOCs [33].

On the other hand, there is still an open question that can be applied to any MOOC: Is there any business model for MOOCs? There is a growing discussion on the MOOC community [45] about sustainability and financial viability (monetization) behind MOOC courses. In order to address this question, it is necessary to make a distinction between objectives of MOOC platforms and course generators. From a MOOC platform perspective, apart from providing an open global learning infrastructure, the main interest lies on attracting the best universities and instructors to publish high-quality courses that can rapidly meet the new skills required in the marketplace. However, it is not just a matter of having a good and up-to-date catalogue of courses, learners must be motivated to get a real and verified certification of their investment (time) and instructors and institutions must see a real return of investment considering the costs of producing each MOOC.

To do so, one of the main actions, promoted by edX and Coursera, was the creation of verified and paid certificates that allowed learners to get an official certificate emitted by the MOOC provider and benefited all stakeholders. Thus, MOOC platforms could attract learners pursuing an official certificate and institutions were also attracted to publish courses and get some revenue due to the economy of scale of reaching thousands of students (even though just a small percentage of them pursued the verified and paid certificate). Beyond issuing single course certificates, edX launched the "XSeries program" [46] that "*cover content equivalent to two to four traditional residential courses and take between six months and two years to complete*". Coursera and Udacity also followed this approach creating "Specializations" and "Nanodegrees" respectively. The main objective of all these new credentials were to indicate some level of competence for high-demand skills and due to this action both platforms raised significant new funding in 2015 to boost more of these credentials: Coursera $61.1 million and Udacity $105 million. Building on this approach, Coursera has recently launched a paywall for graded assignments that conduct to the verified certificate. In this manner, contents are open but not the graded assignments. According to the aforementioned points, it seems clear that courses and contents will be open but the main platforms will discontinue the free honor course certificates looking for getting more engagement and revenue of actions such as the "XSeries program", "Specializations" or "Nanodegrees" that have currently reached up to 100 in 2015.

From the MOOC providers perspective, these actions are actually leading to the concept of on-line universities where institutions will focus on producing high-quality and up-to-date courses and will get revenue for emitting verified certificates. As an example [47], the Johns Hopkins University made at least $3.5 million in less than a year from the sale of verified certificates for its Data Science Specialization. In contrast, HarvardX provided a very good insight in November 2015: they have more than 3 million of enrollments on edX but the revenue coming from verified certificates is only $435,000 (although more than 80% of the HarvardX courses offer verified certificates).

In conclusion, the MOOC platforms are clearly interested in becoming the main channel of on-line education and getting more engagement and revenue from verified and paid certificates of courses and learning pathways such as the "XSeries program", "Specializations" or "Nanodegrees". MOOC providers such as universities will become entities that produce learning contents, design learning pathways and emit verified certificates shifting the traditional degrees to a digital environment. Learners will take advantage of this learning environment to get new and verified skills on-demand. Due to all these facts, on-line education will turn into another on-demand and pay-as-you-go cloud utility service in which contents will be open and free but verified certificates must be paid. However, one of the main criticisms of MOOC still persists: Is it possible to ensure that this on-line learning method actually equip people with the skills that are offered in courses?

5. Conclusions and future work

The sudden rise of MOOCs as disruptors to higher education have also generated a lot of debate around the learning process including evaluation and certification criteria, the lack of skills among instructors to design and plan MOOCs and the technical and security issues. In order to tackle some of these issues, authors propose in this work the application of agile project management principles to guide the development of MOOCs as a

process. A preliminary study of MOOCs as agile projects has been presented to finally make a mapping of agile tasks to agile MOOC activities defining an Agile MOOC Development Lifecycle (AMDL). Afterwards, the AMDL has been applied to the creation of a complete new course in the area of software engineering. More specifically, a case study has been conducted to develop a real MOOC and to validate the presented approach. This course also aims to bring software notions to a large and non-technical audience. People need to be equipped with the required skills to manage software concepts in any knowledge area. To do so, the MOOC has been designed to cover all relevant aspects in the software engineering education encompassing the background of a large multidisciplinary team of instructors (18) to convey the main notions of a wide area such as software engineering to a non-technical audience. The main outcomes of this work is the experience and lessons learned to share with potential MOOC creators the possibilities of applying agile principles to ease the creation of general purpose engineering courses for large and non-technical audiences and to successfully deal with the complexity of managing a large team. Furthermore, an effort estimation has been also outlined to show the need of human and material resources to deliver a complete new MOOC. Future research directions should include the improvement of the learning process and contents, the impact analysis of the course in the community (e.g. dropout rate), combination of the AMDL with existing approaches such as the MOOC Canvas [27–28], study of the return-on-investment as well as the creation of new specific courses in some of the topics presented in this introductory MOOC to fulfill real-world business and technical needs in a software-defined world.

Acknowledgements—This work is supported by the official program "UC3Mx Digital" and the edX platform. We would like to specially acknowledge the commitment and support of all our team members belonging to the Software Engineering Lab (SEL), Knowledge Reuse (KR) and DEI Interactive Systems research groups that have actively participated in the development of this MOOC. Finally, a special acknowledge to the staff of the UTEID (Educational Technology and Innovative Teaching Unit) at UC3M.

References

1. G. Christensen, A. Steinmetz, B. Alcorn, A. Bennett, D. Woods and E. J. Emanuel, The MOOC Phenomenon: Who Takes Massive Open Online Courses and Why?, *SSRN Electron. J.*, 2013.
2. G. Conole, MOOCs as disruptive technologies: strategies for enhancing the learner experience and quality of MOOCs, *RED Rev. Educ. Distancia*, **39**, 2013, pp. 1–17.
3. R. H. Rizzardini, F. J. García-Peñalvo and C. D. Kloos, Massive Open Online Courses: Combining Methodologies and Architecture for a Success Learning, *J-Jucs*, **21**(5), 2015, pp. 636–637.
4. M. Wen, D. Yang and C. P. Rose, Linguistic reflections of student engagement in massive open online courses, in *Proceedings of the International Conference on Weblogs and Social Media*, 2014.
5. D. Leony, P. J. Muñoz-Merino, J. A. Ruipérez-Valiente, A. Pardo and C. D. Kloos, Detection and Evaluation of Emotions in Massive Open Online Courses, *J-Jucs*, **21**(5), 2015, pp. 638–655.
6. D. Yang, T. Sinha, D. Adamson and C. P. Rose, Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses, in *Proceedings of the 2013 NIPS Data-Driven Education Workshop*, Lake Tahoe, NV, USA, 2013, **10**, pp. 13–20.
7. N. Spyropoulou, C. Pierrakeas and A. Kameas, Creating MOOC guidelines based on best practice, in *EDULEARN14 Proceedings*, Barcelona, Spain, 2014, pp. 6981–6990.
8. I. Scott, L. Kendra and A. Woodgate, How to grow your own MOOC with Coursera, University of Edinburgh, 2014.
9. J. Kerr, *Building and Executing MOOCs*, University of Glasgow, 2015.
10. E. Teplechuk, Emergent models of Massive Open Online Courses: an exploration of sustainable practices for MOOC institutions in the context of the launch of MOOCs at the University of Edinburgh, 2013.
11. U. Northwestern, *MOOC Creation Guidelines*, Northwestern University, http://www.northwestern.edu/provost/docs/MOOC_Creation_Guidelines.pdf. Chicago, United States, 2015.
12. U. Toronto, *Massively Open Online Courses (MOOC): Design and Development Process Guidelines*, http://www.ocw.utoronto.ca/wp-content/uploads/2013/05/MOOC-Design-and-Development-Guidelines-2-Feb-2013.pdf. University of Toronto, Toronto, Canada, Feb. 2013.
13. [D. Demaree, A. Kruse, S. Pennestri, J. Russell, T. Schlafly and Y. Vovides, From Planning to Launching MOOCs: Guidelines and Tips from GeorgetownX, in *E-Learning, E-Education, and Online Training*, **138**, G. Vincenti, A. Bucciero and C. Vaz de Carvalho, Eds. Cham: Springer International Publishing, 2014, pp. 68–75.
14. J. Haywood and H. Macleod, "To MOOC or not to MOOC?," *Massive Open Online Courses MOOC Revolut.*, 2014, p. 46.
15. A. Liu, Y. Dang, and S. Lu, Effectiveness of E-learning 2.0 Tools and Services to Support Learner-Learner Virtual Interactions in A Global Engineering Class, *IJEE*, **31**(2), 2015, pp. 553–566.
16. K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, *Principles behind the Agile Manifesto*, Agile Alliance, 2001.
17. S. W. Ambler and M. Lines, *Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise*. Upper Saddle River, NJ: IBM Press, 2012.
18. [A. Begel and N. Nagappan, Usage and perceptions of agile software development in an industrial context: An exploratory study, in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, 2007, pp. 255–264.
19. S. Downes, Week 2: The Quality of Massive Open Online Courses by Stephen Downes.
20. D. Clark, MOOCs: taxonomy of 8 types of MOOC.
21. L. Guàrdia, M. Maina and A. Sangrà, MOOC design principles: A pedagogical approach from the learner's perspective, *ELearning Pap.*, (33), 2013.
22. G. Siemens, Connectivism: A learning theory for the digital age, 2005.
23. J. Mackness, M. Waite, G. Roberts and E. Lovegrove, Learning in a small, task–oriented, connectivist MOOC: Pedagogical issues and implications for higher education, *Int. Rev. Res. Open Distrib. Learn.*, **14**(4), 2013.
24. P. McAndrew, Learning from open design: running a learning design MOOC, *ELearning Pap.*, (33), 2013.
25. Y. Wang and R. Baker, Content or platform: Why do students complete MOOCs?, *J. Online Learn. Teach.*, **11**(1), p. 17, 2015.

26. S. Zheng, M. B. Rosson, P. C. Shih and J. M. Carroll, Understanding student motivation, behaviors and perceptions in MOOCs, in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 1882–1895.
27. C. Alario-Hoyos, M. Pérez-Sanagustín, D. Cormier and C. D. Kloos, Proposal for a Conceptual Framework for Educators to Describe and Design MOOCs, *J UCS*, **20**(1), 2014, pp. 6–23.
28. C. Alario-Hoyos, M. Pérez-Sanagustín, C. D. Kloos and P. J. Muñoz-Merino, Recommendations for the design and deployment of MOOCs: insights about the MOOC digital education of the future deployed in MiríadaX, 2014, pp. 403–408.
29. A. Osterwalder, Y. Pigneur and T. Clark, *Business model generation: a handbook for visionaries, game changers, and challengers*. Hoboken, NJ: Wiley, 2010.
30. Á. Fidalgo-Blanco, M. L. Sein-Echaluce and F. J. García-Peñalvo, Methodological Approach and Technological Framework to break the current limitations of MOOC model, *J. Univers. Comput. Sci.*, **21**, 2015, pp. 712–734.
31. V. Clinton M. and R. David M., Building Leaders: A National Examination of the Leadership Capacities within Engineering Undergraduate Student, *IJEE*, **31**(4), 2015, pp. 986–997.
32. V. Cok, N. Fain, N. Vukasinovic and R. Zavbi, Multicultural issues of product development education in virtual teams, *IJEE*, **31**(3), 2015, pp. 863–873.
33. F. M. Hollands and D. Tirthali, Resource requirements and costs of developing and delivering MOOCs, *Int. Rev. Res. Open Distrib. Learn.*, **15**(5), 2014.
34. O. Skrypnyk, S. Joksimović, V. Kovanović, D. Gašević and S. Dawson, Roles of course facilitators, learners, and technology in the flow of information of a cMOOC, *Br. J. Educ. Technol. Spec. Issue MOOCs*, 2015.
35. M. Bernhart, T. Grechenig, J. Hetzl and W. Zuser, Dimensions of software engineering course design, in *Proceedings of the 28th International Conference on Software Engineering*, 2006, pp. 667–672.
36. B. Dasarathy, K. Sullivan, D. C. Schmidt, D. H. Fisher and A. Porter, The past, present, and future of MOOCs and their relevance to software engineering, in *Proceedings of the on Future of Software Engineering*, 2014, pp. 212–224.
37. NESSI, *Software Engineering-Key Enabler for Innovation*, Networked European Software and Services Initiative, White paper NESSI_SE_WhitePaper-FINAL, 2014.
38. A. D. Lantada, P. L. Morgado, J. M. Muñoz-Guijosa, J. E. Otero and J. L. M. Sanz, Learning through play in a final year subject: enjoyable design experience for teaching product development, *Int. J. Eng. Educ.*, **27**(3), 2011, pp. 488–497.
39. M. A. Ardis and P. B. Henderson, Software engineering education (SEEd): is software engineering ready for MOOCs?, *ACM SIGSOFT Softw. Eng. Notes*, **37**(5), Sep. 2012, p. 14.
40. K. F. Colvin, J. Champaign, A. Liu, Q. Zhou, C. Fredericks and D. E. Pritchard, Learning in an introductory physics MOOC: All cohorts learn equally, including an on-campus class, *Int. Rev. Res. Open Distrib. Learn.*, **15**(4), 2014.
41. F. M. Hollands and D. Tirthali, MOOCs: Expectations and reality, *Cent. Benefit-Cost Stud. Educ. Teach. Coll. Columbia Univ.*, 2014.
42. A. Koutropoulos and P. Zaharias, Down the Rabbit Hole: An initial typology of issues around the development of MOOCs, *Curr. Issues Emerg. ELearning*, **2**(1), 2015, p. 4.
43. U. Carlos III, *MOOCs at UC3M*, Carlos III University. http://www.uc3m.es/ss/Satellite/UC3MInstitucional/en/TextoMixta/1371210164552/MOOCs Accessed: 27-Feb-2016.
44. U. Carlos III, *Calls for Proposal: UC3M Digital and Innovation in Teaching.* Carlos III University. http://www.uc3m.es/ss/Satellite/UC3MInstitucional/en/TextoMixta/1371214349882/. Accessed: 27-Feb-2016.
45. D. Shah, *MOOC Trends in 2015: Big MOOC Providers Find their Business Models*, https://www.class-central.com/report/mooc-business-model/ Accessed: 27-Feb-2016.
46. S. Carson, MITx introduces 'XSeries' course-sequence certificates on edX, *MIT News*, http://news.mit.edu/2013/mitx-introduces-xseries-course-sequence-certificates-on-edx Accessed: 27-Feb-2016.
47. D. Shah, *Eyeing Revenue Sustainability: The Two Biggest MOOC Providers Adapt How Their Courses Work*, https://www.class-central.com/report/coursera-paywall-edx-discontinues-free-certificates/ Accessed: 27-Feb-2016.
48. A. Amescua et al. "The Software Architect Code: Building the Digital World" https://www.edx.org/course/software-architect-code-building-digital-uc3mx-inf-1x Accessed: 27-Feb-2016

Antonio De Amescua, PhD, is a full Professor at Universidad Carlos III de Madrid, Director of the Software Engineering Lab at UC3M. He is a member of ECQA (European Certification and Qualification Association). His research interests focus on Software Process and Learning Organization.

Jose María Alvarez Rodríguez, PhD, is a Visiting Professor at Universidad Carlos III de Madrid. He holds a PhD (2012) about e-Procurement and semantics from the University of Oviedo (Spain). His main research interests are semantic technologies applied to fields such as Systems Engineering or Open Science. He has been granted an HPC2 Europe Transnational Access Program and a Marie Curie Postdoc in Cloud Computing and QoS (Quality of Service). Finally, he is member of international associations and working groups such as INCOSE or OSLC RM (Requirements Management).

María Isabel Sanchez-Segura, PhD, is Associate Professor at Universidad Carlos III de Madrid where she leads the research line of processes and information technology for the governance of intelligent organizations. She is Knowledge Manager Certified by the Knowledge Management Institute and senior member of IEEE (Institute of Electrical and Electronics Engineers).

Fuensanta Medina Domínguez, PhD, is a faculty member of the Computer Science Department at Universidad Carlos III de Madrid since 2004. She holds a Ph.D. in Computer Science from UC3M. Fuensanta has been working in the software engineer field since 2000 and she is Knowledge Manager Certifier by the knowledge Management Institute. Her research interests include software engineering, software process improvement, global software development, Knowledge Management, Intellectual Capital and Intelligent Organizations.