Software Engineering Perspective of Reconfigurable Hardware Programming

Dr. Raafah Khazem[*]
Samer Mumtaz Sulaiman[**]

(Received 21 / 5 / 2019. Accepted 5 / 11 / 2019)

□ ABSTRACT □

Developing a software system is a task that has been done by several companies over a long time, Projects differ in their nature as some require real-time processing, or might have hardware involved in them, so these factors might affect the overall development process and might even have a bad effect if not handled properly. Reconfigurable hardware-based projects are a special case of software projects, which was being concentrated on a lot in Syria in the recent years. The recommendations that were presented to software production companies made a slight improvement to the overall performance of the development team as all these recommendations revolve around the fact that all parts of the team whether they were analysts or programmers must get involved with the hardware being used in order to better understand how to plan and expect this hardware to operate.

Keywords: Software Engineering- Reconfigurable Systems- Software Methodologies- Waterfall Methodology- Agile Methodology- Prototyping Methodology- Custom made methodologies.

[*] **Assistant Professor - Faculty of Mechanical & Electrical Engineering, Department of Computer & Automation Engineering, Damascus University, Damascus, Syria.**

[**] **Postgraduate Student (Master) - Faculty of Mechanical & Electrical Engineering, Department of Computer & Automation Engineering, Damascus University, Damascus, Syria.**

منظور هندسة البرمجيات لبرمجة العتاد القابل لإعادة الضبط

د. رأفة معين خازم*

سامر ممتاز سليمان**

(تاريخ الإيداع 21 / 5 / 2019. قُبِل للنشر في 5 / 11 / 2019)

□ ملخّص □

تقوم العديد من الشركات بتطوير النظم البرمجية بشكل دوري منذ فترة زمنية طويلة، تختلف المشاريع بطبيعتها حيث أنَّ بعض هذه المشاريع قد يتطلب معالجة في الزمن الحقيقي (Real-time) أو قد تتضمن عتادًا كأحد مكونات النظام. فهذه العوامل وغيرها قد تؤثر بعملية التطوير الكلية ويمكن أن يكون لها آثار سلبيّة إذ تمّ التعامل معها بشكل غير مناسب. تعتبر المشاريع البرمجية المعتمدة على العتاد القابل لإعادة الضبط (Reconfigurable hardware-based projects) حالة خاصة من المشاريع البرمجية والتي لقيت اهتمامًا كبيراً في سوريا في السنين الأخيرة. لقد ساهمت التوصيات المقدمة للشركات المنتجة للبرمجيات في تحسين الأداء الكلي لفريق التطوير حيث أنَّ كل هذه التوصيات تدور حول أهمية انخراط فريق التطوير كاملًا من محللين ومبرمجين في العتاد مما يسهم في تحقيق فهم أفضل للمشروع وبالتالي معرفة كيفية التخطيط وما يمكن أن نتوقعه من العتاد بشكل صحيح.

الكلمات المفتاحية: هندسة البرمجيات- الأنظمة القابلة لإعادة الضبط- منهجيات التطوير البرمجية- منهجية الشلال- منهجية التطوير الرشيق- منهجية النمذجة الأولية- المنهجيات المطورة محليًا.

* **مدرسة - كلية الهندسة الميكانيكية والكهربائية، قسم هندسة الحواسيب والأتمتة، جامعة دمشق، دمشق، سورية.**

** **طالب دراسات عليا (ماجستير) ، كلية الهندسة الميكانيكية والكهربائية، قسم هندسة الحواسيب والأتمتة)- جامعة دمشق، دمشق، سورية.**

مقدمة:

تعتبر الحوسبة القابلة لإعادة الضبط(Reconfigurable computing) موضوعًا هامًّا له العديد من التطبيقات في مختلف المجالات، فمع الطلب المتزايد على تحسين قدرة المعالجة، أصبح دمج العتاد القابل لإعادة البرمجة مع وحدات المعالجة أمرًا مرغوبًا. إذ تُعرّف الحوسبة القابلة لإعادة الضبط بأنّها دمج معمارية الحاسب التقليدية مع نسيج قابل لإعادة الضبط الذي يكون عادة مصفوفة البوابات القابلة لإعادة البرمجة حقليًّا Programmable Gate Field Array (FPGA) ويمكن أن يتم استخدام أي نوع من نوع العتاد الآخر القابل للضبط[2]. يتّسم النّسيج القابل لإعادة الضبط بالعديد من المزايا أهمها قدرة المعالجة واستهلاك الطاقة بالإضافة إلى الديناميكية العالية في أداء الوظائف إذ أنه يمكن توظيف هذا العتاد في أداء وظائف مختلفة عبر إعادة برمجته عندما تكون اللوحة في حقل العمل كما أن بعض اللوحات الحديثة تدعم ميزة إعادة الضبط الجزئي الديناميكي Dynamic partial re-configuration والتي تسمح بإعادة ضبط اللوحة لأداء وظيفة مختلفة وهي تتابع مهماتها الحالية دون توقُّف.

يعرّف معهد مهندسي الكهرباء والالكترونيات IEEE هندسة البرمجيات على أنها " التطبيق الممنهج للمعرفة والطرائق والخبرات العلمية والتقنية في تصميم وتحقيق واختبار وتوثيق البرمجيات" حيث نجد أنّ هذا التعريف العام يشمل في تفاصيله حالة برمجة العتاد القابل لإعادة البرمجية إذ أنّ هذا البرنامج الناتج في النهاية هو عبارة عن مجموعة تعليمات سيتم تنفيذها على العتاد سواء كان حاسباً تقليدياً أو نسيج قابل لإعادة الضبط تمَّ إلحاقه بالحاسب. تعدُّ عملية تطوير مثل هذه البرامج مختلفًة عن تطوير البرامج التقليدية سواء في مرحلة التحليل أو التصميم أو التنجيز. يجب قياس مدى ملائمة منهجيات التطوير التقليدية في تطوير النظم القابلة لإعادة الضبط حيث " إنّ اختيار المنهجية المناسبة ليس بالأمر البسيط لأنه لا يوجد هناك منهجية تكون دومًا الأفضل في كل مكان وزمان"[1].

أهمية البحث وأهدافه:

نظراً للتحسين الكبير الذي تُضفيه إضافة النسيج القابل لإعادة الضبط لنظم الحوسبة التقليدية يُمكِن أنْ نعتبر أنّ دراسة النُّظم القابلة لإعادة الضبط من وجهة نظر هندسة البرمجيات خطوة هامة في سبيل تحسين طريقة تطوير هذه النظم بشكل أمثلي وبالتالي إضفاء الجودة لكل من المنتج النهائي وعملية تطويره. يهدف البحث بشكل أساسي إلى دراسة منهجيات التطوير المستخدمة في هندسة البرمجيات التي أصبحت عديدة ومعرفة أي منها يناسب تطوير النظم القابلة لإعادة الضبط.

طرائق البحث ومواده:

تتضمن الدراسة التواصل مع الشركات التي تُعنى بتطوير النظم البرمجية في سوريا، وقد تم التركيز على الشركات التي تنفذ مشاريع برمجية لها طابع عتادي ولا يعني ذلك فقط المشاريع التي تتضمن أي شكل من أشكال العتاد بل تم التركيز على العتاد الذي يمكن إعادة ضبطه أثناء العمل. تمت دراسة أربع منهجيات أساسية في التطوير البرمجي وهي منهجية الشلال (Waterfall)، منهجية النمذجة الأولية (Prototype)، منهجية التطوير الرشيق (Agile) ومنهجية Lean. تمت هذه الدراسة في الفترة الزمنية الممتدة بين 2018/05/15 و 2019/03/30 إذ تمت دراسة عشرين

مشروعاً برمجياً بدأت عقب البدء بالدراسة الحالية وذلك استنادًا إلى القانون الأول في هندسة البرمجيات[4] والذي ينص على أنّ قضايا هندسة البرمجيات يجب أنْ تُدرس من الجانبين النظري والتجريبي على حد سواء.

منهجيات التطوير المدروسة:

يزداد عدد منهجيات التطوير التي يمكن استخدامها في تطوير المشاريع البرمجية بشكل كبير ويكون لكل منهجية خصائص مميزة تجعلها مناسبة لبعض أنواع المشاريع دون الأخرى. ندرس فيما يلي أهم المنهجيات المؤثرة في المشاريع التي تتضمن العتاد القابل لإعادة الضبط:

منهجية الشلال: تعتبر هذه المنهجية من أبسط المنهجيات كما أنها من أقدم المنهجيات التي تم استخدامها في هندسة البرمجيات إذ يمكن تعقبها إلى عام 1956م حيث تم وصفها بأنها تسلسل خطي من المراحل التي يتم اتباعها للحصول على منتج برمجي ما وقد تم استخدام مصطلح الشلال لأول مرة في عام1976م[3]. على الرغم من بساطة هذه المنهجية التي يبدأ المشروع بها بجمع المتطلبات ومن ثم ينتقل إلى التحليل فالتصميم فالتنجيز ومن ثم الاختبار إلا أنها تجعل من الأمر الصعب توقُّع مخرجات النظام وخصوصًا أنّ هذه المراحل قد تستغرق وقتاً طويلًا وهو الأمر الذي تتميز به المنهجيات الأخرى مقارنة بهذه المنهجية.

منهجية النمذجة الأولية: تتميز هذه المنهجية بانخراط المستخدم بشكل كبير في عملية التطوير، إذ يبدأ المشروع بجمع المتطلبات ومن ثم إنشاء النموذج الأولي الذي يتم عرضه على المستخدم الذي يقيم هذا النموذج ويوضح مواضع الخلل فيه ويتم تعديل النموذج وفقاً لذلك عددًا من المرات حتى الوصول إلى النموذج النهائي الذي يتم بناء النظام وفقًا له. نلحظ في هذه الطريقة أن عملية الحصول على النموذج النهائي قد تستهلك وقتًا كبيرًا وذلك يتعلق بخبرة فريق التطوير والمستخدمين على حد سواء ولعل من أهم عيوب هذه المنهجية "عدم ملاءمتها لتطوير النظم المعقدة كون التعرف على المشاكل والمواضيع الأساسية لا يتم حتى فترة متقدمة من إجرائية التطوير"[1].

منهجية التطوير الرشيق: تتسم هذه المنهجية بالتكرارية إذ أن عمليتي التطوير والاختبار تحدثان بشكل مستمر على عكس المنهجيات الخطية كالشلال. فناتج هذه المنهجية قابّل للتغيير بشكلٍ دائم كما يجب التركيز على توثيق هذه التغييرات مهما كانت لأن غيابه سيجعل من الصيانة أمرًا صعبًا كما تزداد صعوبة التخطيط للمرحلة التكرارية المقبلة في ظل غياب توثيق ما تم تغييره مع كل دورة تكرار مما قد يسبب ضياعًا في الموارد المخصصة لعملية التطوير وهو ما يحصل عادة عند استخدام مثل هذه المنهجيات وخصيصًا بوجود فريق تطوير ذي خبرة محدودة.

منهجية Lean: توضح دراسة هذه المنهجية أنها ليست جديدة بل إنها محاولة جدية لأمثلة عمل منهجيات التطوير الرشيق إذ أنها تعتمد على دعامتين في عملها وهما التحسين المستمر و احترام الأشخاص. تركز هذه المنهجية بدعامتيها على مبدأ التخلص من فضلات المشاريع والتي تعرّف بأنها كل ما لايفيد في عملية التطوير ويجب توضيح أنه ليس ذلك هو الهدف الأساسي للمنهجية بل خلق القيمة المضافة للزبون هو الهدف الرئيسي التي تحاول هذه المنهجية الوصول إليه وتستخدم هذه المنهجية لتحقيق ذلك الكثير من التقنيات التي تسهم في تطبيق التحسين المستمر واحترام الأشخاص في المؤسسة مما يوضح أنّ هذه المنهجية لا تعطي النتائج المرجوة منها إلا عند تطبيقها على شريحة كبيرة في المؤسسة العاملة على إنشاء المشروع ولهذا السبب عادة ما تكون نتائجها واضحة إذ تم تطبيقها بشكل صحيح.

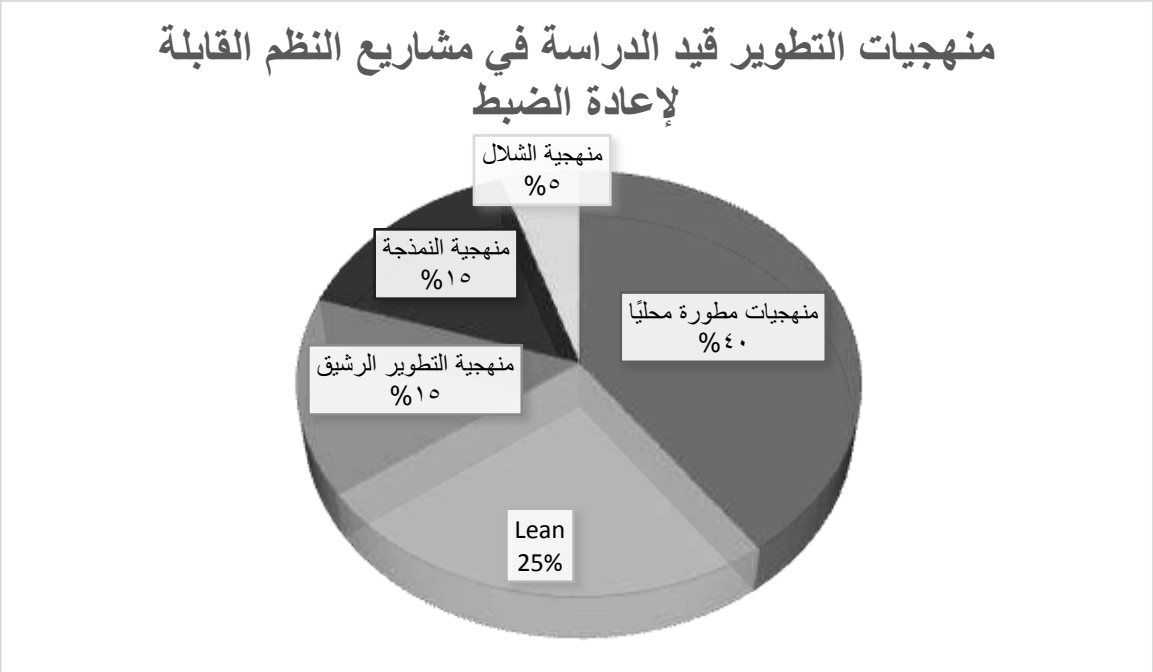
خصوصية تطوير النظم القابلة لإعادة الضبط:

تشترك النظم القابلة لإعادة الضبط مع النظم البرمجية في المتطلبات الأساسية عند التطوير، فمشاريع التطوير لكلا النوعين تطلب تسليم المنتج النهائي للمستخدم بأسرع وقت ممكن وأن ينال رضا الزبون في عملية التطوير وعند استخدام المنتج ودعمه بعد التسليم. عند دراسة مشاريع تطوير النظم القابلة لإعادة الضبط يتضح أنها تضيف بعد الخصوصية إلى مشاريع التطوير التقليدية ويتضح ذلك فيما يلي:

1. ضرورة اطلاع كامل فريق التطوير على مواصفات النسيج القابل لإعادة الضبط، وأن يكتسب الفريق من محللين ومصممين ومبرمجين معرفة كافية عن قدرات هذا العتاد وكيفية ترابطه مع مكونات النظم الأخرى بشكل يزيل أي فهم خاطئ لقدرات العتاد المستخدم ومحدودياته علمًا أنّ هذه المعرفة لا يجب أن تكون محصورة فقط بالمبرمج على عكس ما هو شائع عادة، بل يجب أن يشترك فريق التطوير ككل باختيار صنف النسيج القابل لإعادة الضبط الذي يتناسب بشكل أساسي مع متطلبات المستخدم ويجب أن يقوم محلل النظام بدوره في توضيح النقاط المتعلقة بقدرات العتاد للمستخدم وذلك لإزالة أية توقعات غير منطقية من هذا العتاد والتي تنشأ عادة بسبب ارتفاع كلفة هذا العتاد.
2. تكرار عملية تنصيب الجزء البرمجي المتعلق بالنسيج القابل لإعادة الضبط عدد كبير من المرات أثناء العمل يفرض أنْ تكون أجزاء النظام ككل أو أجزاء النظام التي تتفاعل مع هذا النسيج مترابطة بشكل متراخي مما قد يسمح بإزالة أجزاء من النظام دون أن تتأثر الأجزاء الأخرى.
3. صعوبة إنتاج المشاريع العامة التي تلائم العتاد المصنع من مختلف الشركات المصنعة إذ أنّ النظام يُبنى على أساس نوع معين من النسيج القابل لإعادة الضبط وإنّ استبداله بنسيج آخر تم تصنيعه من قبل شركة أخرى قد يسبب فشل النظام أو عمله بشكل خاطئ حتى ولو كانت مواصفات الجزء الجديد المستخدم أفضل من الذي قد تم استخدامه عند تطوير المشروع. حيث أنّ الشركة التي تقوم بتصنيع المنتج هي الضامن الوحيد لتوافق منتجاتها. هذا الأمر يجعل من النظام محدودًا من ناحية المحمولية كما يفرض قيودًا واضحة على كل من شركة التطوير وعلى الشركة التي ستستخدم المنتج التي يجب أن تدرس خياراتها في حال عدم وجود بديل مناسب للنسيج المستخدم.
4. ضرورة إضافة مرحلة اختبار جديدة تعرف باسم " اختبار قابلية إعادة الضبط (Reconfigurability test)" والتي تدرس مدى إمكانية عمل النظام بشكل صحيح بعد كل عملية إعادة ضبط والتي تضمن إمكانية الوصول إلى حالة آمنة للنظام في حال حدث أي خطأ في عملية إعادة الضبط.

النتائج والمناقشة:

إن النقاط السابقة التي توضح خصوصية النظم القابلة لإعادة الضبط توضح التوصيات التي يمكن تقديمها لأية شركة تعمل على تطوير مثل هذه النظم. وبدراسة حالة شركات تطوير البرمجيات في سوريا فيوضح الشكل(1) نسبة استخدام كل من المنهجيات التي تمت دراستها مسبقاً



الشكل رقم (1):نسبة استخدام منهجيات التطوير في مشاريع تطوير النظم القابلة لإعادة الضبط

يتضح من المخطط أنّ أغلب الشركات لا تلتزم بالشكل الرسمي لمنهجية محددة بل تعتمد منهجيات مطورة من قبلها، وأما النسبة القليلة نسبيًا التي تعتمد منهجيات الرسمية فقد عزت خيارها إلى سببين رئيسيين هما:

1. طبيعة المشاريع التي تلقتها.
2. طبيعة فريق التطوير.

ويتضح من نتائج الاستبيانات التي قدمتها الشركات أن طبيعة فريق التطوير هي السبب الرئيسي في اختيار منهجية التطوير ويتضح ذلك من وجود منهجية الشلال بين المنهجيات قيد الاستخدام والتي أُعزي سبب استخدامها إلى وجود أعضاء في فريق التطوير ممن يكملون دراسة شاملة ودقيقة لكل عناصر النظام قبل البدء بتطويره مما جعل هذه المنهجية الأنسب من بين المنهجيات المدروسة ، فلو عمل هذا الفريق وفقاً لمنهجية ذات طبيعة تكرارية لكانت النتائج سيئة من حيث زمن التسليم. بسبب بساطتها الشديدة تفشل المنهجيات التقليدية مثل الشلال أو منهجية الانفجار الكبير (Big Bang methodology) عادة في مثل هذه المشاريع إلا في بعض الحالات الخاصة آنفة الذكر أما المنهجيات المتبقية من المنهجيات المدروسة فتعد مناسبة لمشاريع النظم القابلة لإعادة الضبط ويوضح الجدول رقم(1) أهم الخصائص التي تميز المنهجيات المدروسة والتي تم التركيز عليها لدورها الكبير عند دراسة النظم القابلة لإعادة الضبط وتعطي منهجية Lean بأدواتها المختلفة أفضل أداء ممكن مقارنة بالمنهجيات الأخرى حيث أنّ معظم المنهجيات المطورة محليًا من قبل الشركات تتشابه بشكل كبير مع هذه المنهجية.

الجدول رقم (1): الخصائص الأساسية للمنهجيات قيد الدراسة في ظل دراسةالنظم القابلة لإعادة الضبط

	الشلال	النمذجة الأولية	التطوير الرشيق	Lean
التعقيد	بسيطة	متوسطة التعقيد	معقدة	معقدة
سرعة تسليم المنتج	بطيئة جدًا	بطيئة	سريعة	سريعة
الفعالية	منخفضة	منخفضة	عالية	عالية و تميل نحو الأمثلة
المحمولية	فقط إذا تم أخذها بعين الاعتبار قيد التطوير وخصيصًا منذ بداية المشروع	فقط إذا تم أخذها بعين الاعتبار قيد التطوير وخصيصًا منذ بداية المشروع	يمكن تحقيقها بسهولة	يمكن تحقيقها بسهولة
قابلية إعادة التنصيب بعد تعديل النظام	ممكن ولكن يستغرق فترة زمنية طويلة قد تصل إلى ستة أشهر لأن التعديل قد يتطلب دورة تطوير مصغرة	ممكن ولكن يستغرق فترة زمنية تتراوح بين الشهر و ستة أشهر مع الأخذ بعين الاعتبار جودة التطوير وعمل فريق التطوير	ممكن ولا تستغرق زمناً طويلاً بفضل التحسين المستمر ويجري ذلك عادة خلال شهر واحد كحد أقصى	ممكن ويستغرق أقل زمن ممكن بفضل التحسين المستمر ويتم عادة خلال عدة أيام.

الاستنتاجات والتوصيات:

تشير نتائج الدراسة التطبيقية إلى أن استخدام المنهجيات التقليدية في تطوير النظم القابلة لإعادة الضبط هو أمر ممكن، وذلك مع الأخذ بعين الاعتبار خصوصية هذه المشاريع ولعل انخراط فريق التطوير كاملاً بدراسة العتاد ومعرفة إمكانياته بالحد الأدنى بقصد إزالة أي تصورات غير منطقية أكانت من طرف فريق التطوير أم من طرف المستخدم النهائي.

إنّ دراسة العتاد القابل لإعادة الضبط من وجهة نظر هندسة البرمجيات هي الخطوة الأولى في الحصول على نظم قابلة لإعادة الضبط ذات جودة عالية تتناسب مع متطلبات المستخدمين وليست الغاية من ذلك فقط استخدام هندسة البرمجيات في هذا المجال من الدراسة فحسب بل الغاية هي إثبات قابلية تطبيق مبادئ هندسة البرمجيات المختلفة بتطوير مختلف النظم فعلى سبيل المثال مع ازدياد الاهتمام بانترنت الأشياء (Internet of Things) يجب أن تتم دراسة هذه الحالة الخاصة من وجهة نظر هندسة البرمجيات أيضاً إذ تتشابه هذه الحالة مع النظم القابلة لإعادة الضبط في الكثير من قيود العمل إذ أنّ عمل تجهيزات انترنت الأشياء يقتضي وجود عتاد خاص وغالبًا ما يكون قابلًا لإعادة الضبط وذلك في بعض وظائفه على الأقل إذ أنّ العتاد القابل لإعادة الضبط ليس بالضرورة أن يتم استخدامه في التشغيل فقط بل يمكن أيضًا أن تتم إضافته من أجل الاختبار والتحقق من سلامة تطوير المنتج سواء كان ذلك من أجل اختبار التصميم أو اختبار التنجيز[5]. وفي تطبيقات أخرى للعتاد القابل لإعادة الضبط نجد نظام التشغيل ReconOs[6] الذي يهدف بشكل أساسي إلى تقديم لغة تخاطب موحدة بين أنواع النسج القابلة لإعادة الضبط مما يوجد حلولًا لمشاكل التوفقية التي تم التنويه لها مسبقًا إلا أنه يطرح الكثير من المشاكل الجديدة إذ يتم بواسطة هذا النظام نقل النياسب البرمجية ليتم تنفيذها بشكل عتادي على النسيج القابل لإعادة الضبط مما يتطلب إضافة وحدات لتنفيذ هذه العملية بنجاح وضمان تقديم جميع الخدمات التقليدية المتعلقة بالنياسب سواء كانت بشكلها العتادي أم البرمجي. يؤكد كل ما سبق على أهمية دراسة كيفية اندماج العتاد القابل لإعادة الضبط بالنظام البرمجي مع التركيز على آلية التفاعل بين العتاد وبرمجته وبين الجزء البرمجي المتبقي من النظام.

References:

(1) KHAZEM, R., System Analysis and Design, 1st edition , publish of Damascus University, Syria,2015, 351.
(2) COMPTON, K. , HAUCK, S. *Reconfigurable Computing: A Survey of Systems and Software*, ACM Computing Surveys, Vol. 34, N°. 2, 2002, 171–210.
(3) BELL, T. , Thayer T., *Software requirements: Are they really a problem?* Proceedings of the 2nd international conference on Software engineering. IEEE Computer Society Press, 1976.
(4) WANG, Y., *Software engineering foundations: A software science perspective*, Auerbach Publications, Auerbach, 2008 ,1419
(5) PATRIGEO, G., LELOUP P., BENOIT, P. TORRES , *FlexNode a reconfigurable Internet of Things node for design evaluation*, France, SophiaAntipolis, 2019.
(6) AGNE, A. *et al*., *ReconOS: An Operating System Approach for Reconfigurable Computing*, in *IEEE Micro*, Vol. 34, N°. 1, 2014 , 60-71.