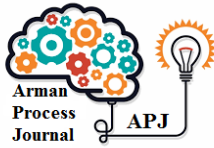


APJ. 4(4):11-18, Winter 2023



## Arman Process Journal (APJ)

Homepage: [armanprocessjournal.ir](http://armanprocessjournal.ir)

### Examining the challenges of software development in an agile way, focusing on non-functional requirements

*M. Yazdanpanah*

*Imam Hussain officer and guard training university, Tehran, Iran*

#### ABSTRACT

Received: 19 December 2023

Accepted: 10 March 2024

#### KEYWORDS:

- . Software Development
- . Non-Functional-Requirements
- . Quality Components
- . Agile Method

<sup>1</sup> Corresponding author



[m.yazdanpanah313@gmail.com](mailto:m.yazdanpanah313@gmail.com)

**Background and Objectives:** The challenges of software development using the agile method, focusing on non-functional requirements, is one of the most important issues in the field of software engineering. Non-functional requirements include things like efficiency, security, usability, maintainability and developability, which are very important in software design and development. However, in the agile method, there are challenges to integrate non-functional requirements with functional requirements, and it is also important to review and evaluate these requirements during the software development process. Non-functional requirements and requirements (NFR) are often reported as a problematic issue. NFRs are qualitative components that focus on expectations and actually help us to have an optimal and easy-to-use system that has Acceptable performance and efficiency. In the investigations carried out, insufficient knowledge, low competence of employees, especially in terms of their analytical skills and professional experience, have been identified among the identified components of neglecting NFR. Our goal in this article is to identify the challenges and problems that affect the implementation of NFRs and the methods that facilitate the successful implementation of NFRs. In order to find the right answer to the questions raised and the right way of working, we have reviewed and summarized the raised literature systematically with a qualitative approach. In the reviewed studies, there was no comprehensive analysis of the techniques used in agile requirements engineering to deal with the challenges of non-functional requirements. In this article, the challenges of software development in the agile method were examined by focusing on non-functional requirements. Non-functional requirements are an important research area that mainly occurs due to the frequency of project failures caused by ignoring the qualitative characteristics related to user values. In the investigations carried out, the increase in development cost and time, the decrease in system flexibility, the increase in complexity and the problems of maintenance and subsequent development, and finally the increase in security risks and software errors, including the damage of neglecting non-functional requirements, have been identified. Effective coordination and cooperation between development groups and other stakeholders have also been identified. It is very important. By following the best practices and non-functional requirements management methods, you can have a significant improvement in the software development process in an agile way with higher accuracy and quality. The study field of software engineering is considered as one of the attractive and practical fields. Undoubtedly, the research conducted in this field will expand day by day and will provide better results. Among the proposed fields for further relevant studies in this field, we can mention the examination of agile management models and methods and improvement of development processes.



NUMBER OF REFERENCES

47



NUMBER OF FIGURES

0



NUMBER OF TABLES

0

## بررسی چالش‌های توسعه نرم‌افزار به روش چابک با تمرکز بر الزامات غیر عملکردی

مرتضی یزدان پناه

دانشگاه افسری و تربیت پاسداری امام حسین علیه السلام، تهران، ایران

### چکیده

**پیشینه و اهداف:** چالش‌های توسعه نرم‌افزار با استفاده از روش چابک با تمرکز بر الزامات غیر عملکردی، یکی از مسائل مهم در حوزه مهندسی نرم‌افزار است. الزامات غیر عملکردی شامل مواردی مانند کارایی، امنیت، قابلیت استفاده، قابلیت نگهداری و قابلیت توسعه هستند که در طراحی و توسعه نرم‌افزار بسیار حائز اهمیت هستند. با وجود این، در روش چابک، چالش‌هایی برای ادغام الزامات غیر عملکردی با الزامات عملکردی وجود دارد و همچنین بررسی و ارزیابی این الزامات در طول فرآیند توسعه نرم‌افزار بسیار مهم است. نیازمندی‌ها و الزامات غیر عملکردی (NFR) اغلب به عنوان یک مسئله مشکل ساز گزارش می‌شود. NFRها مولفه‌های کیفی می‌باشند که بر روی انتظارات متمرکز می‌شوند و در حقیقت برای داشتن سیستم کاربری بهینه و آسان به ما کمک می‌کنند که دارای کارایی و بازده قابل قبولی باشد در بررسی‌های به عمل آمده دانش ناکافی، شایستگی پایین کارکنان به ویژه از نظر مهارت‌های تحلیلی و تجربه حرفه‌ای آنها از جمله مولفه‌های شناسایی شده نادیده انگاری NFR شناسایی شده است. هدف ما در این مقاله شناسایی چالش‌ها و مشکلاتی است که بر اجرای NFRها تأثیر می‌گذارد و شیوه‌هایی که اجرای موفقیت آمیز NFRها را تسهیل می‌کنند. به منظور یافتن پاسخ مناسب به پرسش‌های مطرح شده و راه کار مناسب ما با یک رویکرد کیفی مروری بر ادبیات مطرح شده بطور نظام مند را بررسی و جمع‌بندی کرده ایم. در مطالعات بررسی شده تحلیل جامعی که از تکنیک‌های مورد استفاده مهندسی نیازمندی به روش چابک برای مقابله با چالش‌های نیازمندی‌های غیر عملکردی صحبت به میان آورده باشد مشاهده نگردید. در این مقاله، چالش‌های توسعه نرم‌افزار به روش چابک با تمرکز بر الزامات غیر عملکردی مورد بررسی قرار گرفت. نیازمندی‌های غیر عملکردی یک حوزه تحقیقاتی با اهمیت می‌باشد که عمدتاً به دلیل فراوانی شکست‌های پروژه ناشی از نادیده گرفتن ویژگی‌های کیفی مرتبط با ارزش‌های کاربر رخ می‌دهد. در بررسی‌های انجام شده افزایش هزینه و زمان توسعه، کاهش انعطاف‌پذیری سیستم، افزایش پیچیدگی و مشکلات نگهداری و توسعه بعدی و در نهایت افزایش خطرات امنیتی و خطاهای نرم‌افزاری از جمله آسیب‌های بی‌توجهی به الزامات غیر عملکردی شناسایی شده است. هماهنگی و همکاری مؤثر بین گروه‌های توسعه و سایر ذینفعان نیز بسیار حائز اهمیت است. با رعایت بهترین شیوه‌ها و روش‌های مدیریت الزامات غیر عملکردی، می‌توان بهبود قابل توجهی در فرآیند توسعه نرم‌افزار به روش چابک با دقت و کیفیت بالاتر داشت. حوزه مطالعاتی مهندسی نرم‌افزار از جمله حوزه‌های جذاب و کاربردی محسوب می‌شود بدون شک روزه‌روز تحقیقات انجام شده در این زمینه گسترش پیدا کرده و نتایج بهتری را ارائه خواهد کرد. از جمله زمینه‌های پیشنهادی برای مطالعات مرتبط بعدی در این زمینه می‌توان به بررسی مدل‌ها و روش‌های مدیریت چابک و بهبود فرآیندهای توسعه اشاره کرد.

### واژگان کلیدی:

توسعه نرم‌افزار  
نیازمندی‌های غیر عملکردی  
مولفه‌های کیفی  
روش چابک

## مقدمه

توسعه نرم‌افزار چابک یکی از رویکردهای ارائه محصولات نرم‌افزاری است. اصطلاح چابک به معنای سازگاری [۱]، انعطاف‌پذیری و همکاری نزدیک با مشتری یا کاربر نهایی [۲] [۳] می‌باشد. رویکردهای روش چابک ارزش‌هایی مانند اعتماد [۴] مسئولیت و وفاداری و غیره را در نظر می‌گیرد. در حال حاضر بسیاری از سازمان‌ها از شیوه‌های چابک برای اتخاذ مدیریت تغییر و تحول استفاده می‌کنند [۵]، [۶].

علاوه بر این، نتایج یک نظرسنجی انجام‌شده در سال ۲۰۱۸ در میان متصدیان حوزه نرم‌افزار نشان می‌دهد که ۹۷ درصد از پاسخ‌دهندگان از روش‌های چابک استفاده کرده‌اند. [۷] در واقع، مزایای استفاده از شیوه‌های چابک در بسیاری از مطالعات و پژوهش‌های انجام‌شده گزارش شده است که نشان‌دهنده افزایش بهره‌وری گروه [۲]، انگیزه و نظم و همچنین کیفیت کلی نرم‌افزار است [۸]، [۹]. کیفیت نرم‌افزار جنبه بسیار مهمی است که باید در طول چرخه عمر نرم‌افزار در نظر گرفته شود [۱۰]، [۱۱] که معمولاً بر اساس ویژگی‌های سطح بالا تعریف می‌شود [۱۲]. FR یا نیازمندی‌های عملکردی، در واقع نیازمندی‌هایی است که کارفرما برای سامانه یا نرم‌افزاری که قصد پیاده‌سازی و تولید آن را دارد، صراحتاً به گروه پروژه اعلام می‌کند. این نیازمندی‌ها مشخصاً قابلیت‌هایی است که یک نرم‌افزار باید داشته باشد تا اهداف کارفرما را محقق سازد و کارفرما از آن شناخت کافی دارد. در مقابل NFR یا همان نیازمندی‌های غیر عملکردی می‌گوید برای هر کدام از این نیازها، چه مواردی باید پیش‌بینی شود تا نرم‌افزار بتواند نیازمندی‌های کارفرما را به‌درستی پوشش دهد. به‌عنوان مثال کارفرما سامانه‌ای نیاز دارد تا دانشجویان بتوانند درخواست وام خود را به‌صورت آنلاین ثبت کنند و روند اداری آن را پیگیری نمایند. گروه پروژه باید بتواند شرایطی ایجاد کند که در روزهای پیک ثبت‌نام، سرویس‌دهی سامانه مطلوب باشد و با کندی مواجه نشود به عبارتی کارایی داشته باشد.

از طرف دیگر، سامانه امن باشد و اطلاعات دانشجویان توسط افراد متفرقه قابل مشاهده و دست‌کاری نباشد. نیازمندی عملکردی در حقیقت عملی که سیستم باید انجام دهد را توصیف می‌کند. نیازمندی غیر عملکردی نحوه عملکرد سیستم و چگونگی رفتار آن و همچنین محدودیت‌های آن را تعیین می‌کند. در حقیقت NFR نیازمندی‌هایی را نشان می‌دهد که توسط نیازمندی‌های عملکردی FR پوشش داده نشده است. از ابتدای توسعه نرم‌افزار، NFR ها به‌عنوان عوامل مهمی شناخته شده‌اند که بر پذیرش و استفاده از محصولات توسط کاربران تأثیر می‌گذارند [۱۳] برای کاهش نارضایتی کاربران با سوءتفاهم یا نادیده گرفتن انتظارات و نیازهای آن‌ها، مشارکت فعال کاربر در توسعه نرم‌افزار امری ضروری و اجتناب‌ناپذیر است [۱۴]، [۱۵] ارزش‌ها و اصولی که در توسعه نرم‌افزار به روش چابک دنبال می‌شوند، منجر به شیوه‌هایی متفاوت از روش‌های سنتی توسعه نرم‌افزار می‌شوند. این موارد شامل شیوه‌های مهندسی نیازمندی‌ها است [۱۶]، [۱۷].

## نیازمندی‌های غیر عملکردی NFR

مهندسی نیازمندی‌ها اساساً فرآیندی کل‌نگر است که هدف آن دریافت نیازمندی‌ها و وابستگی‌های سیستمی می‌باشد، بنابراین به‌سختی می‌توان یک فرآیند مجزا برای یک دسته نیازمندی‌های خاص مانند NFRها طراحی کرد.

باین‌حال، هنوز هم می‌توان تکنیک‌ها و شیوه‌های مختلفی را در فرآیند مهندسی نیازمندی‌ها به کار برد، از جمله تکنیک‌های اختصاصی توسعه‌یافته برای مقابله با دسته‌بندی‌های خاصی از نیازمندی‌ها. نیازمندی‌های غیر عملکردی که به‌عنوان کیفیت نیز شناخته می‌شوند، انتظارات و نیازهای کاربران را در رابطه با محصولات نرم‌افزاری تعریف می‌کنند. مهم‌ترین ویژگی‌های کیفیت شامل قابلیت استفاده، عملکرد، قابلیت اطمینان، پایداری، ایمنی، امنیت، یکپارچگی، انطباق، قابلیت نگهداری، قابلیت استفاده مجدد و قابلیت همکاری می‌باشد. بدون تردید، NFR ها اهمیت زیادی در توسعه محصولات نرم‌افزاری دارند [۱۸]، [۱۹]. NFRها همچنین می‌توانند محدودیت‌های کلانی را بر یک محصول نرم‌افزار تحمیل کنند که ناشی از بخش‌ها و وابستگی بین آن‌ها می‌باشد [۲۰] [21] به‌عبارت‌دیگر، NFR ها محدودیت‌هایی را در مورد نحوه عملکرد محصول ایجاد می‌کنند. نادیده گرفتن اطلاعات مربوط به جنبه‌های کیفی بر محصول نهایی تأثیر منفی می‌گذارد. از این رو، ادعا می‌شود که خطاهای مربوط به NFR سخت‌ترین و گران‌ترین اصلاحات را دارا می‌باشند [22, 23].

مایکروسافت که بزرگ‌ترین شرکت نرم‌افزاری در سراسر جهان است، دریافت نیازهای کاربردی و غیر عملکردی را با اولویت یکسان توصیه می‌کند، زیرا نیازمندی‌های کاربردی نشان می‌دهد که آیا برنامه کار را درست انجام می‌دهد یا خیر، درحالی‌که نیازمندی‌های غیر عملکردی تعیین می‌کند که آیا برنامه آن کارها را به‌خوبی انجام می‌دهد یا خیر. Oracle نیز که از جمله شرکت‌های بزرگ نرم‌افزاری می‌باشد، استدلال می‌کند که "کلید توسعه نرم‌افزار موفق این است که همه ذینفعان درک واضح و یکنواختی از الزامات برنامه ایجاد کنند" و تعارضی در میان نباشد. چنین چالش‌هایی می‌تواند منجر به یک مانع بزرگ در پذیرش فناوری و استفاده بیشتر از آن شود، زیرا مثال‌های متعددی وجود دارد که نشان می‌دهد کیفیت ناکافی سیستم منجر به کنار گذاشتن آن توسط کاربران نهایی می‌شود [۱۲] - [۱۴] و NFR ها راه اصلی بیان انتظارات و مشخصه‌های کیفیتی هستند.

[۲۴]، [۲۵]. علاوه بر این موارد اهمیت NFR ها به‌عنوان جنبه‌های اصلی کیفیت محصولات نرم‌افزاری از دیدگاه کاربر بر کسی پوشیده نیست [۲۶]. نیازمندی‌های غیر عملکردی به یک حوزه تحقیقاتی مهم تبدیل شده با توجه به اینکه عمدتاً شکست‌هایی که در پروژه‌ها ناشی می‌شود مربوط به نادیده گرفتن ویژگی‌های مربوط به کیفیت و ارزش‌های کاربر می‌باشد. درحالی‌که

و همکاران بر معرفی دیدگاه کاربر به توسعه نرم‌افزار و کاربرد مشترک توسعه و طراحی کاربر محور تمرکز دارد. آن‌ها یک مطالعه برای خلاصه کردن شیوه‌های مرتبط و شناسایی جنبه‌های ضروری مهندسی نیازمندی‌های چابک در زمینه ی طراحی کاربر محور انجام دادند [۳۸].

تعداد پروژه‌های صنعتی که درس‌های متفاوت و خاصی را در مورد NFR ارائه می‌دهند، مقایسه مطالعات را به کاری زمان‌بر و پیچیده تبدیل می‌کند، زیرا آن‌ها اغلب با تمرکز یا اهداف یکسانی سروکار ندارند. گذر کردن از NFR ها دشوارتر است و مشکلاتی را هم برای ذینفعان و هم برای گروه پروژه ایجاد می‌کند. حتی ممکن است ذینفعان نیازهای خود را که باید به‌عنوان NFR در نظر گرفته شوند، تشخیص ندهند. ممکن است گروه پروژه به‌نوبه خود فاقد دانش و شایستگی‌های لازم برای شناسایی و پیاده‌سازی برخی از NFRها باشد، به‌ویژه زمانی که مفاهیم پیشرفته مرتبط با مواردی مانند امنیت یا عملکرد باید استفاده شود [۱۵].

چالش "غفلت از NFR ها در حین تمرکز بر عملکرد" [۲۸] [۳۰] موضوعی مربوط به فعالیت‌های استخراج نیازمندی‌ها است و نشان‌دهنده کاستی‌های احتمالی در تکنیک‌ها/عملکردهای استخراجی مورد استفاده یا استفاده نامناسب از آن‌ها است. برخی منابع [۲۹, ۳۹] گزارش می‌دهند که این چالش ناشی از نادیده گرفتن NFR ها یا برخورد سطحی با آن‌ها در مراحل اولیه/تکرار پروژه است، بنابراین نشان می‌دهد که زمان استخراج NFR ها تکنیک‌های بکار گرفته‌شده حیاتی است و بنابراین ممکن است ارزش بررسی را داشته باشد. چالش‌های نقل‌شده باقی‌مانده [29] و "ناکافی بودن تکنیک‌های موجود نیازمندی‌ها برای مقابله با چالش‌های NFRها" هر دو به تکنیک‌های مستندسازی نیازمندی‌ها مربوط می‌شوند [۳۴].

تکنیک‌های مهندسی نیازمندی‌های مورد استفاده در پروژه‌های Agile شیوه‌ها چالش‌های مهندسی نیازمندی‌ها، کیفیت چابکی الزامات و زمینه‌های تحقیقاتی اصلی در ARE محورهای مطالعه تحقیقاتی مورد توجه ما را پوشش می‌دهند [۳۸, ۴۰]. استخراج و ارتباط NFR ها دسته دیگری از مسائل است. استخراج الزامات می‌تواند همه ذینفعان مربوطه را درگیر نکند و منجر به الزاماتی شود که همه دیدگاه‌ها را منعکس نمی‌کنند یا حتی برخی الزامات را پوشش نمی‌دهد.

بیان NFRها نیز بسیار سخت است، بنابراین ارتباطات آن‌ها (هم از طرف ذینفع به گروه پروژه و هم بین اعضای گروه) ممکن است مستعد خطا باشد. علاوه بر این، در پروژه‌های توسعه در مقیاس بزرگ، که شامل گروه‌های متعدد است، احتمالاً مشکلات ارتباطی اضافی ایجاد می‌شود. نیازمندی‌های غیر عملکردی یک حوزه تحقیقاتی مهم می‌باشد، که عمدتاً به دلیل فراوانی شکست‌های

هیچ اتفاق نظری در مورد دلایل این امر وجود ندارد، Maxim و Kessentini خاطر نشان می‌کنند که NFRها "برای ذینفعان به راحتی قابل بیان نیستند اما آن‌ها می‌دانند که نرم‌افزار بدون برخی از این ویژگی‌های غیر عملکردی قابل استفاده نخواهد بود". علاوه بر این، زمانی که فرد موفق به نوشتن نیازمندی‌ها می‌شود، هنگام تلاش برای مستندسازی ویژگی‌های خاص و وابستگی‌ها و روابط آن‌ها با مشکلاتی مواجه می‌شود [۲۷].

مهندسی نیازمندی‌های به روش چابک مشکلات مرتبط با نیازمندی‌ها را کاهش می‌دهد، به‌عنوان مثال. مقابله با نیازهای متغیر [۲۸] و ارتباط ناکافی با مشتری [۲۹]، اما در عین حال چالش‌های خاص خود را نیز معرفی می‌کند [۳۰, ۳۱]. علیرغم چالش‌های مرتبط با NFR که اغلب ذکر شده است، گزارش‌هایی وجود دارد که متخصصان اهمیت NFRها را در پروژه‌های چابک تشخیص می‌دهند [۳۲, ۳۳] و ادعا می‌شود که نسبت موفقیت پروژه‌های چابک بیشتر از پروژه‌هایی است که از رویکرد برنامه محور استفاده می‌کنند. این می‌تواند نشان دهد که در پروژه‌های چابک، متخصصان فرآیندهای مهندسی شده مناسبی را ایجاد می‌کنند که آن‌ها را قادر می‌سازد به‌طور مؤثر NFRها را ضبط کنند و سپس یک محصول نرم‌افزاری را فراهم کنند که چنین نیازهایی را برآورده کند.

با وجود این مشاهدات، یک سؤال مطرح می‌شود: چرا NFR تا نادیده گرفته شده‌اند؟ دلیل اول، دانش ناکافی و شایستگی پایین کارکنان به‌ویژه از نظر مهارت‌های تحلیلی و تجربه حرفه‌ای آن‌ها است. علاوه بر این، با نادیده گرفتن منابع NFRها تعاریف مبهم و توصیفات مبهمی نشان داده می‌شود. Alsuaquaf و همکاران، NFR ها را در زمینه ی خاصی از پروژه‌های توزیع شده در مقیاس بزرگ توسعه چابک در نظر گرفته‌اند. این نویسندگان یک مطالعه برای خلاصه کردن چالش‌ها و راهکارها انجام دادند [۳۴] و موارد را از طریق یک سری مصاحبه با متخصصان بررسی کردند.

Behutiye [۳۵] و همکاران NFR ها را در زمینه عمومی توسعه نرم‌افزار در نظر گرفته‌اند. آن‌ها به تجزیه و تحلیل روش‌ها و موقعیت‌هایی برای شیوه‌های مدیریت NFR و همچنین انجام مصاحبه‌هایی برای شناسایی چالش‌ها و مستندسازی راهکارها پرداخته‌اند [۳۶, ۳۷]. Heikilla و همکاران از طریق یک مطالعه نقشه‌برداری، مزایای مهندسی نیازمندی چابک و همچنین نقاط مشکل‌ساز آن را بررسی و تجزیه و تحلیل کرد [۲۸] Medeiros و همکاران یک مطالعه نقشه‌برداری با تمرکز بر روی شیوه‌های مهندسی نیازمندی چابک و تکنیک‌های اختصاص داده شده به استخراج نیازمندی‌ها و مستندسازی آن‌ها انجام داد [۳۱]. Schon.

الگریانی و کاما یک مرور ادبیات سیستماتیک در مورد شیوه‌ها و چالش‌های ARE را مبسوط بررسی گزارش کردند. بررسی آن‌ها مبتنی بر NFR نبود و بنابراین نمی‌توان تعیین کرد که اقدامات گزارش شده تا چه حد برای NFR ها اعمال می‌شود [45].

مقاله‌ای از Zamudio و همکاران در مورد تکنیک‌های مهندسی نیازمندی‌ها را گزارش می‌دهد. نویسندگان بر تکنیک‌های استخراج شده در روش‌های چابک به‌طور خاص تمرکز می‌کنند. چنین تکنیک‌هایی به‌هیچ‌وجه با توجه به کاربرد و اثربخشی آن‌ها در استخراج NFR ها تجزیه و تحلیل نمی‌شوند. در بخش پایانی، نویسندگان بیان می‌کنند که «برای درک بهتر تأثیر شیوه‌های مهندسی نیازمندی‌های چابک، به نتایج تجربی بیشتری نیاز است.» [40] یک مرور ادبیات سیستماتیک با تمرکز اولیه بر NFR ها توسط Alsquaf و همکاران انجام شد. هدف آن خلاصه کردن شیوه‌های مورد استفاده برای مهندسی نیازمندی‌ها در پروژه‌های توزیع شده در مقیاس بزرگ بود، همچنین شناسایی چالش‌های NFR و راه‌حلی برای مقابله با چالش‌های آن نادیده گرفته شده است [۳۴]. هک و زیدمن نتایج به‌دست آمده را بر اساس مشخصات مورد نیاز چابک ارائه کردند. هدف آن‌ها شناسایی معیارهای کیفی چنین مشخصات ذکر شده در ادبیات پژوهش بود. یکی از معیارهای شناسایی شده به گنجاندن NFR ها در مشخصات می‌پردازد. توصیه‌های مرتبط بیان می‌کند که NFR ها نباید نادیده گرفته شوند و جلساتی برای بحث در مورد NFR باید در نظر گرفته شود [۴۶].

کورسیو و همکاران یک مطالعه برای شناسایی موضوعات تحقیقاتی مهندسی نیازمندی‌ها به رویکرد چابک را انجام داده‌اند. علاوه بر این، آن‌ها چالش‌ها (موانع) مرتبط با فعالیت‌های ARE را خلاصه کردند. از آنجایی که تمایز خاصی از NFR ها صورت نگرفته است، موضوعات تحقیقاتی فهرست شده به‌عنوان مثال منابع نیازمندی‌ها یا تکنیک‌های استخراج دسته‌های مختلف نیازمندی‌ها را به‌صورت مشترک در نظر می‌گیرند. لیست چالش‌ها مختص NFR نمی‌شود، اما یک شکاف کلی از پشتیبانی ناکافی از NFR در روش‌های چابک گزارش شده است [۴۷].

### نتیجه‌گیری

در این مقاله، چالش‌های توسعه نرم‌افزار به روش چابک با تمرکز بر الزامات غیر عملکردی مورد بررسی قرار گرفت. نیازمندی‌های غیر عملکردی یک حوزه تحقیقاتی با اهمیت می‌باشد که عمدتاً به دلیل فراوانی شکست‌های پروژه ناشی از نادیده گرفتن ویژگی‌های کیفی مرتبط با ارزش‌های کاربر رخ می‌دهد. با توجه به اهمیت موضوع و مغفول ماندن در برخی موارد این سؤال پیش می‌آید که چرا این اتفاق رخ می‌دهد؟ بر اساس بررسی‌ها، اضافه کردن الزامات غیر

پروژه ناشی از نادیده گرفتن ویژگی‌های کیفی مرتبط با ارزش‌های کاربر رخ می‌دهد. درحالی‌که هیچ اتفاق نظری در مورد دلایل این امر وجود ندارد، در برخی مطالعات نیازمندی‌های غیر عملکردی "برای ذینفعان به‌راحتی قابل بیان نیستند، اما آن‌ها می‌دانند که نرم‌افزار بدون برخی از این ویژگی‌های غیر عملکردی قابل استفاده نخواهد بود" [۲۷].

به‌طور معمول چند کلیدواژه در مطالعات قابل مشاهده می‌باشد که مربوط به مشکلات استخراج نیازمندی‌های غیر عملکردی می‌باشد از جمله نادیده گرفتن، فقدان، سوءبرداشت و غیره که حتی زمانی که فرد موفق به نوشتن آن‌ها می‌شود، در طول مسیر در تلاش برای مستندسازی کیفیت‌های خاص و وابستگی‌های آن‌ها با مشکلاتی مواجه می‌شود. دانش ناکافی و شایستگی پایین کارکنان به‌ویژه از نظر مهارت‌های تحلیلی و تجربه حرفه‌ای آن‌ها نیز از جمله مواردی می‌باشد که نیازمندی‌های غیر عملکردی پروژه را تحت تأثیر مستقیم خود قرار می‌دهد. علاوه بر این، تجربه ناکافی آن‌ها نیز با نادیده گرفتن منابع نیازمندی‌های غیر عملکردی، تعاریف و توصیفات را مبهم نشان می‌دهد که این موضوع می‌تواند نیازمندی‌ها را از اصل خود دور کرده و به حاشیه ببرد.

### مهندسی نیازمندی‌های چابک

AGILE به‌طور گسترده توسط فروشندگان نرم‌افزار [۴۱]، سازمان‌ها [۴۲] و بسیاری از مجموعه‌های دیگر برای اشاره به تغییر از روش‌های سنتی به چابک پذیرفته شده است. گزارش‌های ارائه شده در مطالعات و پژوهش‌های به‌عمل آمده نشان می‌دهد که اتخاذ روش‌های چابک منجر به افزایش بهره‌وری و روحیه تیمی، بهینه‌سازی کار و صرف انرژی کمتر و کارآمدتر شدن کار و محیط می‌شود [۴۳، ۴۴]. مهندسی نیازمندی‌های به روش چابک بسیاری از مشکلات مهندسی نیازمندی‌ها را کاهش می‌دهد که همین موضوع توجه محققان را به خود جلب کرده است [۳۰].

مطالعه Medeiros و همکاران [۳۱] بررسی کرد که چگونه مهندسی نیازمندی‌ها در پروژه‌هایی که متدولوژی‌های چابک را اتخاذ می‌کنند، انجام شده است. در میان سؤالات تحقیقاتی خاص‌تر مطرح شده توسط این مطالعه، «تکنیک‌های استخراج نیازمندی‌ها»، «تکنیک‌های مشخص‌سازی نیازمندی‌ها» و «چالش‌های مرتبط با آن» را می‌توان یافت. باین‌حال، یافته‌های آن‌ها مشخص نمی‌کند که از کدام تکنیک‌ها برای استخراج/مستند کردن NFR استفاده می‌شود. فهرست طولانی چالش‌های شناسایی شده تنها شامل یک چالش مرتبط با NFR است که بیان می‌کند که تکنیک‌های مهندسی نیازمندی به روش چابک در تعریف NFR ضعیف هستند.

- [6] Roy, S., A. Raju, and S. Mandal, An empirical investigation on e-retailer agility, customer satisfaction, commitment and loyalty. *Business: Theory and Practice*, 2017. 18: p. 97-108.
- [7] Diebold, P. and U. Mayer. On the usage and benefits of agile methods & practices. in *International Conference on Agile Software Development*. 2017. Springer, Cham.
- [8] Bjarnason, E., K. Wnuk, and B. Regnell. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. in *proceedings of the 1st workshop on agile requirements engineering*. 2011.
- [9] Kaur, K. and A. Jajoo. Applying agile methodologies in industry projects: Benefits and challenges. in *2015 International Conference on Computing Communication Control and Automation*. 2015. IEEE.
- [10] Guzmán, L., et al. How can quality awareness support rapid software development?—a research preview. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2017. Springer.
- [11] Guaman, D.S., J.M. Del Alamo, and J.C. Caiza, A systematic mapping study on software quality control techniques for assessing privacy in information systems. *IEEE access*, 2020. 8: p. 74808-74833.
- [12] Jarzębowski, A. and K. Połocka. Selecting requirements documentation techniques for software projects: a survey study. in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2017. IEEE.
- [13] Ryan, A.J. An approach to quantitative non-functional requirements in software development. in *Systems engineering—a key to competitive advantage for all industries: proceedings of the 2nd european systems engineering conference (eusec 2000)*, munich. 2000.
- [14] Kautz, K. Customer and user involvement in agile software development. in *International Conference on Agile Processes and Extreme Programming in Software Engineering*. 2009. Springer.
- [15] Jarzębowski, A. and P. Marciniak, A survey on identifying and addressing business analysis problems. *Foundations of Computing and Decision Sciences*, 2017. 42: p. 315-337.
- [16] Jarzębowski, A. and W. Ślesiński. What is troubling IT analysts? A survey report from Poland on requirements-related problems. in *KKIO Software Engineering Conference*. 2018. Springer.
- [17] Leffingwell, D., *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. 2010: Addison-Wesley Professional.
- [18] Svensson, R.B., et al., Quality requirements in industrial practice—an extended interview study at eleven companies. *IEEE transactions on software engineering*, 2011. 38(4): p. 923-935.
- [19] Zhang, X. and X. Wang, Tradeoff analysis for conflicting software non-functional requirements. *IEEE Access*, 2019. 7: p. 156463-156475.
- [20] Weichbroth, P., Delivering usability in IT products: empirical lessons from the field. *International Journal of Software Engineering and Knowledge Engineering*, 2018. 28(07): p. 1027-1045.
- عملکردی به فرآیند توسعه به صورت چابک، می تواند چالش هایی را در پیش رو قرار دهد. برای مثال، باید توجه شود که چگونه الزامات غیر عملکردی در فرآیند توسعه محاسبه شود و چگونه آن ها را به صورت مستمر در فرآیند توسعه اعمال کنیم. علاوه بر این، نیاز به تعادل بین سرعت و انعطاف و نیاز به اجرای الزامات غیر عملکردی نیز وجود دارد. در نتیجه، باید از ابزارها و روش های مناسب برای مدیریت الزامات غیر عملکردی استفاده کنیم. همچنین، هماهنگی و همکاری مؤثر بین گروه های توسعه و سایر ذینفعان نیز بسیار حائز اهمیت است. با رعایت بهترین شیوه ها و روش های مدیریت الزامات غیر عملکردی، می توان بهبود قابل توجهی در فرآیند توسعه نرم افزار به روش چابک با دقت و کیفیت بالاتر داشت.
- مطالعات و جمع بندی های انجام شده تحقیق را به این سو هدایت می کند که دانش ناکافی و شایستگی پایین کارکنان به ویژه از نظر مهارت های تحلیلی و تجربه حرفه ای آن ها می تواند نتیجه پروژه را تحت الشعاع خود قرار دهد و گاهی منجر به شکست نیز بشود. علاوه بر این، عدم وجود ادبیات و تعاریف مشترک نیز از جمله مواردی است که می توان به آن اشاره کرد. در نتیجه بررسی های انجام شده در تحقیقات صورت گرفته حاکی از آن است که بی توجهی به الزامات غیر عملکردی می تواند باعث افزایش هزینه و زمان توسعه، کاهش انعطاف پذیری سیستم، افزایش پیچیدگی و مشکلات نگهداری و توسعه بعدی گردد و در نهایت به افزایش خطرات امنیتی و خطاهای نرم افزاری ختم شود. حوزه مطالعاتی مهندسی نرم افزار از جمله حوزه های جذاب و کاربردی محسوب می شود. بدون شک روز به روز تحقیقات انجام شده در این زمینه گسترش پیدا کرده و نتایج بهتری را ارائه خواهد کرد. از جمله زمینه های پیشنهادی برای مطالعات مرتبط بعدی در این زمینه می توان به بررسی مدل ها و روش های مدیریت چابک و بهبود فرآیندهای توسعه به روش چابک اشاره کرد.

## منابع و مآخذ

- [1] Amjad, S., et al., Calculating completeness of agile scope in scaled agile development. *IEEE Access*, 2017. 6: p. 5822-5847.
- [2] Adnan, M. and M. Afzal, Ontology based multiagent effort estimation system for scrum agile method. *IEEE Access*, 2017. 5: p. 2.۵۹۹۳-۲۶۰۰۵
- [3] Strandberg, P.E., et al., Information flow in software testing—an interview study with embedded software engineering practitioners. *IEEE Access*, 2019. 7: p. 46434-46453.
- [4] Tjørnehøj, G., M. Fransgård, and S. Skalkam. Trust in agile teams in distributed software development. in *Proceedings of the 35th Information Systems Research Seminar in Scandinavia*. 2012.
- [5] Martin, R.C., *The principles, patterns, and practices of agile software development*. 2002, Prentice Hall.

- scale distributed agile: An empirical study. *Information and software technology*, 2019. 110: p. 39-55.
- [36] Behutiye, W., et al. Non-functional requirements documentation in agile software development: challenges and solution proposal. in *International conference on product-focused software process improvement*. 2017. Springer.
- [37] López, L., et al. Agile quality requirements management best practices portfolio: A situational method engineering approach. in *International Conference on Product-Focused Software Process Improvement*. 2017. Springer.
- [38] Schön, E.-M., et al. Key challenges in agile requirements engineering. in *International Conference on Agile Software Development*. 2017. Springer, Cham.
- [39] Sachdeva, V., Requirements prioritization in agile: use of planning poker for maximizing return on investment, in *Information Technology-New Generations*. 2018, Springer. p. 403-409.
- [40] Zamudio, L., et al. A requirements engineering techniques review in agile software development methods. in *International Conference on Computational Science and Its Applications*. 2017. Springer.
- [41] Kaddoumi, T. and M. Watfa. A proposed agile enterprise architecture framework. in *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*. 2016. IEEE.
- [42] Alhuseini, M.U. and M.M. Olama, 5G service value chain and network slicing framework using ecosystem modeling, agile delivery, and user-story automation .*IEEE Access*, 2019. 7: p. 110856-110873.
- [43] Mitre-Hernández, H.A., et al., Decreasing rework in video games development from a software engineering perspective, in *Trends and Applications in Software Engineering*. 2016, Springer. p. 295-304.
- [44] Kišš, F. and B. Rossi. Agile to lean software development transformation: A systematic literature review. in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2018. IEEE.
- [45] Elghariani, K. and N. Kama. Review on Agile requirements engineering challenges. in *2016 3rd International conference on computer and information sciences (ICCOINS)*. 2016. IEEE.
- [46] Heck, P. and A. Zaidman, A systematic literature review on quality criteria for agile requirements specifications. *Software Quality Journal*, 2018. 26(1): p. 127-160.
- [47] Curcio, K., et al., Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 2018. 139: p. 32-50.
- [21] Suryawanshi, T. and G. Rao, A survey to support NFRs in agile software development process. *Int. J. Comput. Sci. Inf. Technol*, 2015. 6(6): p. 5487-5489.
- [22] Rosa, N.S., G.R. Justo, and P.R. Cunha. A framework for building non-functional software architectures. in *Proceedings of the 2001 ACM symposium on Applied computing*. 2001.
- [23] Mizouni, R. and A. Salah, Towards a framework for estimating system NFRs on behavioral models. *Knowledge-Based Systems*, 2010. 23(7): p. 721-731.
- [24] Hussain, A., E.O. Mkpojiogu, and F.M. Kamal, The role of requirements in the success or failure of software projects. *International Review of Management and Marketing*, 2016. 6(7): p. 306-311.
- [25] Kolf, F. and C. Kerkmann, Lidl software disaster another example of Germany's digital failure. *Handelsblatt Global*, 2018.
- [26] Ossowska, K., et al. Exploring an ontological approach for user requirements elicitation in the design of online virtual agents. in *EuroSymposium on Systems Analysis and Design*. 2016. Springer.
- [27] Maxim, B.R. and M. Kessentini, An introduction to modern software quality assurance, in *Software quality assurance*. 2016, Elsevier. p. 19-46.
- [28] Heikkilä, V.T., et al. A mapping study on requirements engineering in agile software development. in *2015 41st Euromicro conference on software engineering and advanced applications*. 2015. IEEE.
- [29] Ramesh, B., L. Cao, and R. Baskerville, Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 2010. 20(5): p. 449-480.
- [30] Inayat, I., et al., A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, 2015. 51: p. 915-929.
- [31] Medeiros, J., et al. Requirements Engineering in Agile Projects: A Systematic Mapping based in Evidences of Industry. in *CibSE*. 2015.
- [32] Farid, W.M. The Normap methodology: Lightweight engineering of non-functional requirements for agile processes. in *2012 19th Asia-Pacific software engineering conference*. 2012. IEEE.
- [33] Kopczyńska, S., M. Ochodek, and J. Nawrocki, On importance of non-functional requirements in agile software projects—a survey, in *Integrating Research and Practice in Software Engineering*. 2020, Springer. p. 145-1.ΔΔ
- [34] Alsaqaf, W., M. Daneva, and R. Wieringa. Quality requirements in large-scale distributed agile projects—a systematic literature review. in *International working conference on requirements engineering: foundation for software quality*. 2017. Springer.
- [35] Alsaqaf, W., M. Daneva, and R. Wieringa, Quality requirements challenges in the context of large-

---

**COPYRIGHTS**

©2023 by the authors. Published by the Islamic Azad University, Khodabandeh Branch, Zanjan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0>

---

