Advances in Radiation Oncology (2022) 7, 100768

advances in radiation oncology

www.advancesradonc.org

Scientific Article

A Safe and Practical Cycle for Team-Based Development and Implementation of In-House Clinical Software

Jean M. Moran, PhD,[1] Kelly C. Paradis, PhD, Scott W. Hadley, PhD, Martha M. Matuszak, PhD, Charles S. Mayo, PhD, Katherine Woch Naheedy, MS, Xiaoping Chen, PhD, Dale W. Litzenberg, PhD, James Irrer, BE, Maria G. Ditman, MS, Pam Burger, CMD, and Marc L. Kessler, PhD*

Department of Radiation Oncology, Michigan Medicine, University of Michigan, Ann Arbor, Michigan

Received March 11, 2021; accepted July 19, 2021

Abstract

Purpose: Due to a gap in published guidance, we describe our robust cycle of in-house clinical software development and implementation, which has been used for years to facilitate the safe treatment of all patients in our clinics.

Methods and Materials: Our software development and implementation cycle requires clarity in communication, clearly defined roles, thorough commissioning, and regular feedback. Cycle phases include design requirements and use cases, development, physics evaluation testing, clinical evaluation testing, and full clinical release. Software requirements, release notes, test suites, and a commissioning report are created and independently reviewed before clinical use. Software deemed to be high-risk, such as those that are writable to a database, incorporate the use of a formal, team-based hazard analysis. Incident learning is used to both guide initial development and improvements as well as to monitor the safe use of the software.

Results: Our standard process builds in transparency and establishes high expectations in the development and use of custom software to support patient care. Since moving to a commercial planning system platform in 2013, we have applied our team-based software release process to 16 programs related to scripting in the treatment planning system for the clinic.

Conclusions: The principles and methodology described here can be implemented in a range of practice settings regardless of whether or not dedicated resources are available for software development. In addition to teamwork with defined roles, documentation, and use of incident learning, we strongly recommend having a written policy on the process, using phased testing, and incorporating independent oversight and approval before use for patient care. This rigorous process ensures continuous monitoring for and mitigatation of any high risk hazards.

Sources of support: NIH P01 CA059827-20.

Disclosures: Dr Moran, Dr Mayo, and James Irrer report funding from Varian Medical Systems related to this work. Dr Moran reports funding from NIH related to this work and Blue Cross Blue Shield of Michigan unrelated to this work. Dr Matuszak reports funding from Varian Medical Systems and Blue Cross Blue Shield of Michigan unrelated to this work.

Research data are not available from this work.

*Corresponding author: Marc L. Kessler, PhD; E-mail: mkessler@med.umich.edu

[1] Author current institution is Memorial Sloan Kettering Cancer Center (MSKCC), New York, New York.

https://doi.org/10.1016/j.adro.2021.100768

Introduction

In-house-developed software has a long history in radiation oncology. The first 3-dimensional treatment planning systems (TPSs) were created in academic environments.[1-5] More recently, in-house developed software has incorporated automation to enhance the safety and efficiency of patient care, such as for treatment plan creation considering historical planning results[6-9] or plan evaluation.[7,10,11] Other software may support machine quality assurance and workflow improvements.[12]

The basic principles of implementation of commercial hardware and software systems are included in American Association of Physicists in Medicine (AAPM) Task Group 40[13] and, specifically for TPSs, in Task Group 53.[14] The Food and Drug Administration oversees medical device approval, including commercial TPSs, through the 510(k) process.[15] Several TPSs now support application programming interfaces (APIs) to enhance the clinical usability and customization for a wide range of users, including those without formal software development training, to develop and implement custom software for clinical use.[16,17] Overall, guidance is lacking on workflows for the safe development, implementation, and refinement of in-house developed software for clinical use.

Because guidance is lacking, the local team is responsible for defining and providing oversight of any in-house developed software that may affect patient care decisions. Transparency among the interprofessional team is supported with a consistent scope, testing rigor, and existence of documentation for in-house developed software. Such a structure supports quality and safety at the level of the institution, employees, and patients. This is achievable even with the substantial variation among clinical resources directly engaged in development and testing of in-house developed software, ranging from single physicists to interprofessional teams, including software developers.

We have previously reported on a subset of our software to support patient safety[10,12] as well as on how to leverage a fusion of incident learning and failure mode and effects analysis to enable data-driven targeting of high-risk errors.[18]

In this work, we describe our development and implementation cycle for the safe use of in-house developed clinical software. In the Methods and Materials section, we report on how the cycle was modified from its initial development in support of UMPlan, our in-house clinical TPS at the University of Michigan,[19,20] and was then adapted to support use of other clinical software developed in house. We also report on aspects of the process that have supported the success of the cycle, including teamwork, defined roles, formal commissioning, hazard analysis (when appropriate), and an ongoing feedback loop tied to incident reporting. In the Results and Discussion section, specific examples demonstrate how this process has supported clarity in communication in the clinical environment, and this information is compared with other examples in the radiation therapy and medical physics literature. Recommendations are provided to support the variety of resources that may be available for software development and implementation. This software cycle approach has been applied broadly at our institution for both TPS and non-TPS software development. It may be valuable for other institutions that are developing improved tools to support safety and efficiency in patient care.

Methods and Materials

Our department history of innovation includes the development and clinical use of one of the first 3-dimensional TPSs as well as one of the first computer-controlled radiation therapy systems.[19,20] In the early 2000s, we formalized our development, testing, and clinical release processes for our TPS as well as for data transfer from the TPS to a commercial treatment management system (ARIA; Varian, Palo Alto, CA). This team-based approach was motivated by Food and Drug Administration requirements and lessons learned from clinical experiences and industry experts. The process evolved over several years through regular committee meetings of an interdisciplinary team, and through triage meetings with formal, and informal feedback provided from clinical users to physicist stakeholders as new versions of the TPS were developed and released to the clinic. When transitioning to a commercial TPS (Eclipse; Varian, Palo Alto) in 2013, we used the same development, testing, and documentation processes to support software developed using the TPS API and for other software to address safety and/or efficiency gaps. Formal policies were then developed to ensure consistency in the documentation and review process so that the process was independent of personnel. The clinical software development and maintenance cycles are shown in Figure 1.

Teamwork, defined roles, and communication

The pathways for software development are shown in Table 1 and explained here for the different phases. During the concept phase involving our software developer team, a written request is first created by a clinical stakeholder and submitted for review by our team to maximize effective use of this scarce resource. The clinical stakeholder may be any member of the department. When the initial stakeholder is not a physicist, the overall process is then carried forward by a primary physicist stakeholder who gathers input from the departmental committees and/or specific user groups or individuals. Important elements of the request include the goals of the software, potential clinical effect, and usability requirements. A graphic can be helpful to show how the software will be used in the existing workflow. For example, Supplementary Materials - Multimedia File 1 shows a

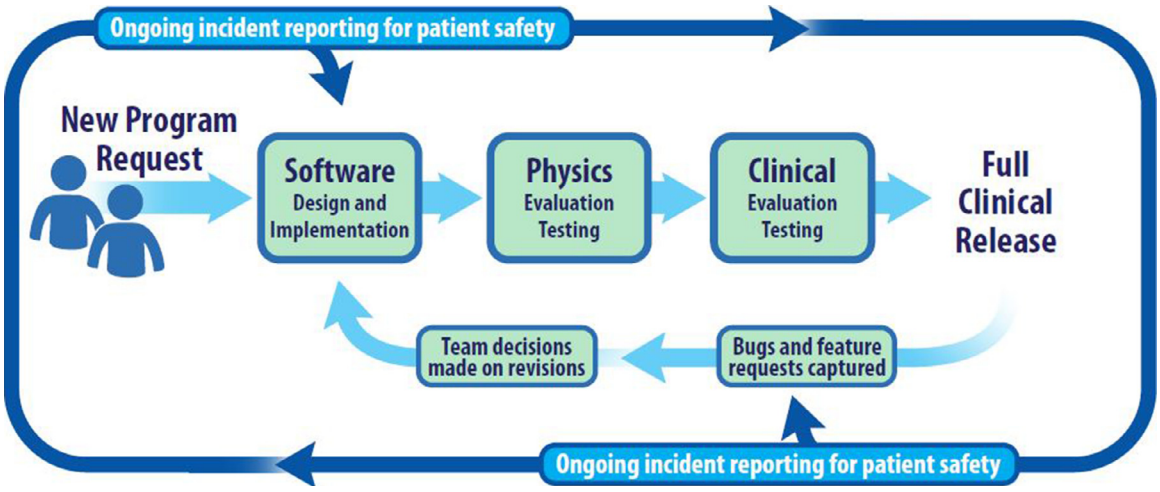

Fig. 1 The software release cycle for new software and maintenance of existing clinical software are shown. Before clinical evaluation testing (CET), a commissioning report must be reviewed and approved.

flow chart of the major stages for the software that ultimately was implemented clinically to support treatment plan checks. Alternatively, when new software development is both conceived of and executed by a single team member, then development is initiated with a prototype, proof-of-concept version. The team is then engaged to discuss the prototype version, review, and make recommendations before initiating our standard software development and release cycle.

Once a request for software developer resources has been submitted, requirements, estimated development effort, and projected clinical effect are assessed by the team. If development is approved, the primary physicist stakeholder duties include writing detailed software requirements for the software developer, specifying user interface needs, providing timely feedback to the software developer during the development process, creating a test suite for commissioning and ongoing quality assurance, leading the commissioning of the software, overseeing user documentation, and training the appropriate users. Due to limited resources, we are mindful of both initial and ongoing development needs. At all stages there is attention to the original scope of the project to monitor requests for extra resources for both development and maintenance. Table 1 shows the roles of the involved team members with an analog presented for clinics without software developers.

A subset of users is trained to provide at-the-elbow support in the clinic. An interprofessional triage team is assembled to monitor the clinical implementation of software and guide decisions related to safety, usability, and priorities for future development. This group participates in deciding if halting clinical deployment is ever necessary. In 2020, we made another improvement by creating a standard communication template for users regarding software changes and possible effects on user groups (Supplementary Materials - Multimedia File 2).

There are 3 main meetings to monitor if there have been any changes in our clinical practice that affect the software. These include bimonthly clinical physics meetings, bimonthly triage meetings with an emphasis on newer software versions in the clinic, and a meeting with software developers and information technology professionals, which includes a clinical physicist liaison.

Commissioning and software release phases

Depending on the role a software tool plays in the treatment plan creation and evaluation process, different levels of commissioning and development are required.

The standard commissioning and testing detailed in Table 1 is sufficient for software tools that will be applied retrospectively; however, increased diligence is needed for higher risk applications, such as those related to prescriptions and plan assessment (including normal tissue complication probability), which may influence treatment decisions. Table 2 details additional testing that should be considered for these higher risk applications. Software that are used for treatment plan creation and evaluation have a higher burden of testing and commissioning because of the associated systematic risk. The work products for these higher risk software tools are specified in Table 2.

With respect to the development and commissioning process, the use of a standard test suite is beneficial when retesting is required due to external changes, such as a new version of the TPS. In these cases, all custom software is tested in a nonclinical version of the patient database, modified as needed, and then is fully commissioned

Table 1 The work product and communication pathways for clinical software development and implementation

Phase	Owner	Work product	Communication pathways (verbal and written) for read-only software
Concept	Clinical stakeholder (typically a physicist)	Software proposal	Oversight is provided from clinical physics and software development. The department's multi-disciplinary leadership team is kept informed of the status of new software.
Software development	Software developer	Initial software version, version control, software engineering release notes	The primary clinical stakeholder provides a set of test cases for development. Once the software is functional, release notes are created for the intended clinical users.
PET	Primary physicist stakeholder	Feedback to the software developer, commissioning report	A primary physicist stakeholder commissions the software and may be supported by a dosimetrist or other stakeholder. The commissioning report is submitted for review and approval. Training is performed emphasizing any items of clinical impact. Once approved, the software is promoted to CET. The application is accessed in a directory tree structure with a folder labeled "PET" at the top to clarify status for end users.
CET	Primary physicist stakeholder	Triage team meetings, collection of bugs and feature requests	Clinical use is allowed with extra scrutiny. A triage team is created to provide support. The team communicates regularly to monitor for critical bugs and future enhancement requests. Training materials are updated if needed. Any incidents related to the software are reported in the department's incident learning system for review and follow-up. Once approved at this stage, the software is promoted to clinical. The application is accessed in a directory tree structure with a folder labeled "CET" at the top to clarify status for end users.
Clinical	Primary physicist stakeholder and director of clinical physics	Stable clinical use of software	Software that reaches this phase has been confirmed to be stable. Software use is monitored. The application is accessed in a directory tree structure with a folder labeled "clinical" at the top to clarify status for end users.

Abbreviations: CET = clinical evaluation testing; PET = physics evaluation testing.

Table 2 Additional process steps for software deemed to be higher risk due to effect on clinical decisions

Stage or step	Owner	Work product	Communication pathways (verbal and written) for software with higher risk regarding patient care
Concept	Primary physicist stakeholder	Software proposal	Some projects may transition from funded research projects toward clinical release and wider input may be gathered at a later stage.
Software development	Software developer	Program	No additional requirements.
Risk assessment	Independent physicist	Detailed hazard analysis	Multiple stakeholders review the software concept and brainstorm possible hazards using the failure mode and effects analysis principles. Hazards are scored based on potential risk to the patient. Proposals for risk mitigation are discussed and documented.
Software development to mitigate hazards	Software developer	Software engineering release notes	The hazard analysis is reviewed in the context of the updated software and updates the hazard analysis. Software may be modified to confirm the application log usage information and capture any errors to support future debugging efforts. Key items are flagged for training.
PET	Primary physicist stakeholder	Feedback to the software developer Commissioning report	Our automated plan check tool[9] reports on any API-based software that has edited a treatment plan or structure set to ensure that such software is not used prematurely. Applications in PET phase cannot be used clinically. Testing may be limited to a small (1-3) group of physicists and dosimetrists who are defined in the TPS with rights to access the application that is approved for evaluation for those in the TPS.
CET	Primary physicist stakeholder	Triage team meetings Collection of bugs and feature requests	For TPS launched software, the application is approved for clinical use. Use is limited, by instruction, to a larger (4-5) specific group of physicists and dosimetrists. The primary physicist stakeholder works with the approved dosimetrists and is responsible for gathering feedback.
Clinical	Primary physicist stakeholder and director of clinical physics	Stable clinical use of software	No additional requirements

Abbreviations: API = application programming interface; CET = clinical evaluation testing; PET = physics evaluation testing; TPS = treatment planning system.

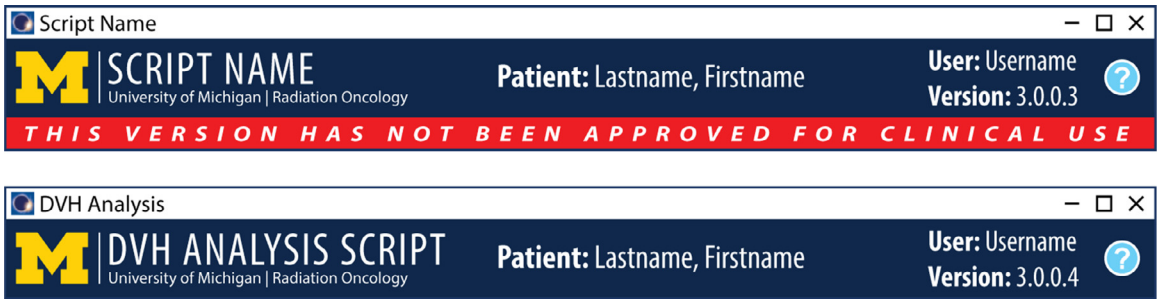

Fig. 2 All software includes a banner with a standard format for (a) preclinical use and (b) after approval for clinical use for an example script. Version numbers distinguish between major and minor releases.

again for clinical use with the standard test suite. Other steps in this stage include using a checklist to confirm the full range of functionality, updating training documents, and reviewing for possible interactions with other software to minimize any unintended consequences of the software release. When there are major changes to the software in this stage, the primary physicist stakeholder will perform testing in the nonclinical system using the relevant, previously designed commissioning tests and will then repeat that testing when the new version is moved to the clinical database.

Once the software is installed for physics evaluation testing, the primary physicist stakeholder assesses the performance of the software with respect to functionality and safety and provides detailed feedback to the developer. Other stakeholders may provide support in this phase. Our software is commissioned following guidance from AAPM Task Group 40[13] and Task Group 53.[14] When applicable, we follow national standards in the presentation of information to users. For example, any software involving prescription information conforms to the essential data elements for a treatment prescription and format as recommended by the American Society for Radiation Oncology white paper by Evans et al.[21]

Basic functionality checks are performed by the software developer, and the primary physicist stakeholder is responsible for testing all software features. It is crucial that the limits of the software are tested, documented, and understood by clinical users. Software-generated error messages are examined to determine whether they provide enough information to investigate the origin of an error. Error messages are incorporated into the documentation and training of all users.

Long-standing programs may be tested by the developer using a batch mode function for an existing test suite designed by the physicist stakeholder or for use on a broader set of predefined patient cases for automated testing on hundreds of data sets and extraction of key results. This saves significant time for both the developer and the primary physicist performing commissioning because the performance of the tests and the test results can be automated for review.

Once commissioning is complete, a report is submitted to the clinical physics lead for review and approval. Other independent reviewers are added if needed and the report is distributed to the team. Training materials are developed as needed and may be customized by user type. Training for major software is documented and is done using standard team or department forums. There may be multiple iterations until the team is ready to recommend approval. Once approved, the software is promoted to the clinical evaluation testing phase (CET).

During the CET phase, clinical rollout is done with a limited number of users and extra scrutiny for those patients. The rollout may be limited in scope by specific team members or treatment sites. The primary physicist stakeholder trains a subset of users to provide clinical support during this phase. Additional feature requests and bug reports are gathered. Triage team meetings are held regularly to discuss clinical flow, stability of the software, and if any showstoppers have been identified that require a halt in clinical use during this phase. If halted, a patch may be made and tested for release directly to CET or the software may be returned to the development phase. If critical bugs are identified, a team will determine whether the software should return to the beginning of the cycle or if the software should be updated. When correcting critical bugs, the software is taken down for the correction to be applied, tested again, and documentation is updated. Users are kept informed of any changes to the status of the software. During this phase, any prior clinically released software is maintained to ensure continuity of clinical operations.

After the CET phase, software is released to the clinical phase. Because the stability of the software was confirmed during CET, any prior clinical software is replaced so that it is the only version available for clinical use until the cycle is repeated. Feature requests and bugs are documented for review by the triage team to assist in setting priorities when it is time for the next phase of

development. Any incidents related to the software are reported in the department's incident learning system for review and follow-up.

The software release cycle in Figure 1 and described in Tables 1 and 2 has been followed for software of different clinical scopes. Independent oversight is a key part of the process. Therefore, commissioning reports are submitted to the director of clinical physics or, if that person is directly involved in the development, to another physicist leader, for review before being approved for the CET phase. We have a dedicated quality and safety officer whose duties include reviewing any reports where there is a safety aspect to the work. After the commissioning report is approved, the CET phase permits clinical use by a subset of users in the clinical environment. Permission settings that restrict software use to customizable user groups are used to support this stage. The users provide feedback directly to the primary physicist stakeholder. The proper functionality of the software has already been confirmed in a test environment, but this extra scrutiny is critical for a safe and successful deployment. When there is an existing clinical version of the software, that version continues to be used to support patient care. During this phase, triage team meetings may be held depending on the software type to ensure the primary physicist stakeholder and software developer are aware of any items requiring attention. Once the software is determined to be robust with clinical use, the user team recommends a transition to full clinical release. The prior clinical version is retired after the new version completes the CET phase. After the clinical stage is stable at our main site, the program is installed at an initial community practice site. An enterprise-wide physicist supports commissioning with local physicists at each of our community practice sites, including transferring anonymized commissioning data sets if applicable, training, and ensuring appropriate documentation at the new clinic. Once it is confirmed as applicable in that new site, it is then implemented and commissioned at other community practices.

All software releases have version labels (eg, 1.1.1.1) to have a distinct record of each version. In the TPS, the last digit is used to indicate the phase (development: 1, physics evaluation testing: 2, CET: 3, clinical: 4). Applications are recompiled with the new version number when advanced and then stored in our directory structure reflecting the phases to maintain clarity. A banner was developed to standardize the presentation of key information, including the status for clinical use for software developed via the API in the TPS. An example of the standard banner is shown in Figure 2. In addition, a determination is made during the design process regarding the method of logging patient records that were accessed using that software. This is invaluable should any problems be identified with the software at a later time.

Although details on software development are outside the scope of this work, it is important to note that certain principles are expected to be followed, such as the use of repositories for storing files, source control management, use of configuration files, and unit testing.[22,23] Software developed for the clinic should be able to be traced and audited.[24]

Prospective and retrospective risk analyses

Our team has extensive experience with using safety and event reporting to drive software development,[10] as well as with incorporating risk analysis techniques such as failure mode and effects analysis[18,25] to enhance patient safety in our clinic. When the scope of our software development expanded beyond read-only software, our team incorporated a hazard analysis into our software release process (see Supplementary Materials - Multimedia File 3 for an example). We updated our formal software release policy, which is overseen by our department interprofessional leadership team, to require a formal hazard analysis in our process for any software that guides clinical decision making, due to the higher clinical risk involved. The hazard analysis is facilitated by an experienced team member. Elements of the analysis include brainstorming by team members with different experience levels and disciplines on potential hazards that could occur when using the software. After brainstorming, hazards are classified by the team as a high, medium, or low risk when considering harm to a patient.[16] A determination is made of whether the hazard was already present in our current workflow (this can help inform mitigation strategies). Any hazards ranked as high risk are identified for a discussion on remediation by the software development team along with the primary clinical stakeholder. Remediation recommendations are then reviewed by a team of physicists and software developers before implementation to confirm the reasonableness of the approach. Hazard mitigation is approved if the approach will result in a decrease of the hazard from a high-risk ranking to medium or low. The hazard analyses for past software implementations have typically started with asynchronous brainstorming led by the facilitator followed by approximately 2 sessions of 1 hour each to do the brainstorming in our software design process.

In-house developed clinical software intersects with patient safety, transparency in our processes, and a culture of safety among our team in 2 important ways. First, we encourage submission of any incidents related to the software. This allows us to improve our process and enhance safety for future patients. Second, many of our software tools support both safety and efficiency and so when reviewing events, individuals are encouraged to consider if any new software or software enhancements can be used to prevent future safety-related events.

As demonstrated in Figure 1, a critical aspect of our in-house software cycle for clinical release is the regular reporting of any bugs or incidents. Any events that involve the use of in-house-developed software are reported in our in-house incident learning system with follow-up analysis reported to the national American Society for Radiation Oncology AAPM Radiation Oncology Incident Learning System.[26] It is crucial that logs are maintained that allow us to identify all patients who may have been affected by an event. For example, our clinic has had 2 situations in which software was used clinically before clinical release. This led to the creation of a different format for the banner for preclinical release software (Fig 2a). Owing to our formal practices and the logging used for our clinical-grade software, any patients who were affected were quickly identified and could be reviewed by our quality committee.

Results and Discussion

Since 2013, we have applied our team-based software release process to 16 programs related to scripting in the TPS for the clinic. It has also been applied to non-TPS-related software related to machine quality assurance and clinical workflow. A sample of our software programs, both TPS and non−TPS-related, is shown in Table 3. Some software programs are launched directly from the TPS whereas others are stand-alone.

For a program to work through this cycle, it is very much dependent on the personnel, the depth of their domain knowledge, the number of projects competing for time, and the availability of the clinical stakeholders and other testers. Small, simple feature enhancements for existing software can take as little as a few days whereas the initial development cycle and preclinical work can take many months for new programs or for a major reconfiguration of an old program. Communication between the developer and the clinical stakeholder and alignment of the timing is important. For example, the process time may be extended if the developer does not receive timely feedback from the clinical stakeholder throughout the process.

As noted in the introduction, radiation oncology has a long history of institutional software development of TPSs in support of quality and patient safety. Among commercial planning systems, Philips Pinnacle TPS was one of the earliest to support development of scripts for automation, which, for example, Purdie et al[27] used to standardize and automate planning for breast cancer. Other commercial TPSs now support automation via APIs. For example, Olsen et al[8] used the Eclipse API to automate contour generation, beams, and treatment plan optimization to support planning. They incorporated a quality control report for the API to provide feedback on whether or not the plan parameters would meet previously approved physician-specified parameters.

In the broader safety perspective, many elements need to come together to have a robust and safe program for patient care.[28,39] AAPM's Task Group 100 report recommends the use of formal risk analysis methods for managing quality in radiation therapy.[30] We have incorporated that philosophy by performing hazard analyses for writable software (software that edits a clinical database). Similarly, Covington et al[31] reported on the development of software to eliminate shift errors in response to incidents reported in their own clinic and in Radiation Oncology Incident Learning System.[32]

Since November of 2017, we have tagged 171 reported events as identifying processes for remediation with further development of a range of our in-house software tools.[10,16] Event reporting is used to help monitor the effectiveness of improvements. After considering Reason's Hierarchy of Effectiveness, which Marks et al[28] helped highlight to the radiation oncology field, we have developed a system that incorporates a clear role for software solutions in addressing safety gaps.

We believe our approach is generalizable to a wide range of clinic sizes, as demonstrated by our deployment at our main site and community practice clinics. With respect to the required resources, Table 4 emphasizes the importance of the documentation types discussed in this report along with examples of how clinics with different levels of resources may be able to incorporate the principles presented in their own clinics. The resources and individuals available to carry out the several roles discussed may vary among clinics. Depending on the resources, the same individual may play several of the roles presented in our work. In all clinical settings, it is important to assure an objective review by more than 1 person with the necessary domain knowledge and to have clear communication across the clinic to support the full cycle, including clinical release. We acknowledge that there are limitations of this work regarding transfer of our methodology to other clinics, especially in light of the decades-long experience by this team in developing and following the cycle described here. We have attempted to give guidance for situations in which limited resources are available to support such a workflow. We believe that independence in software performance evaluation is critical, as is a just culture that supports open and honest discussion of any software-related safety events.

There are other considerations that are outside the scope of the current work related to data integrity (Integrating the Healthcare Enterprise - Radiation Oncology and Digital Imaging and Communications in Medicine efforts); data transfer where applicable, such as addressed in the AAPM Task Group 201[33]; usability; security concerns[34]; and information assurance.

Table 3 A sample of in-house developed software, both TPS and non-TPS, which have been deployed for clinical use

Software name	Year released	Type	Important features
Winston-Lutz image analysis	2011	Automated analysis; report uploaded	Analysis of SRS pretreatment linear accelerator QA; documentation[12]
Analysis of DVH	2015	Read-only	Calculates and reports DVH metrics for any contoured structure (dose-based, volume-based, and NTCP where model available) Supports collation of information in a standard format for reirradiation special medical physics consults[35]
UM Export Agent	2015	Automated data push; no edits to data	AutomaticDICOM-RT pull from TPS, conversion to proper dose and bin settings, and push to 3rd party second check software[12]
Complexity Metric for IMRT[36]	2015	Read-only	Quantifies delivery complexity for IMRT beams and VMAT arcs and compares to previously used clinical plans that passed rigorous pretreatment measurement-based quality assurance
External beam plan check software[10]	2015	Read-only	Tool with automation of key safety feature checks for dosimetrist after plan creation and physicist to support the plan check. All information is collated in 1 document.
Linear Accelerator Scheduling software	2017	Read-only	Reports allowable treatment machines based on plan criteria (such as energy, MLC type) and patient characteristics (such as weight and need for anesthesia)
Blueprint (Planning directive and prescription software)	2018	Read from TPS; writes to own database	Writable software used by physicians, dosimetrists, and physicists to document physician intent, prescription, and record QA checks. Defines datasets and volumes for contouring and evaluates dosimetric plan goals and IGRT instructions. Used in a read-only mode to guide treatment for therapists and to display alerts.
Plan Comparison software	2018*	Read-only	Compares key plan and beam information such as dose per fraction, monitor units, positions for gantry, collimator, jaws, and individual control points if applicable[12]
SRS Metric Check	2018	Read-only	Calculates plan quality measures for a standardized planning approach for multi-target SRS
Brachytherapy plan check tool[37]	2019	Read-only	Evaluates key information for a brachytherapy plan prior to treatment, generates a second check for TG-43
Autoplanning for SRS	2020	Creates structures and plans	Automated generation of structures for optimization, PTV- and OAR-driven isocenter placement, creation of treatment plan and beams, optimization, and dose calculation.
Autoplanning for Head and Neck	2020	Creates structures and plans	Automated generation of structures for optimization, PTV- and OAR-driven isocenter placement, creation of treatment plan and beams, optimization and dose calculation. Algorithmically created optimization constraints combining statistical analysis of historic treatments with current user-defined PTV-OAR trade-offs.

Abbreviations: AAPM = American Association of Physicists in Medicine; DVH = dose volume histogram; DICOM-RT = Digital Imaging and Communications in Medicine - Radiation Therapy; IGRT = image-guided radiation therapy; IMRT = intensity-modulated radiation therapy; MLC = multileaf collimator; NTCP = normal tissue complication probability; OAR = organ at risk; PTV = planning target volume; QA = quality assurance; SRS = stereotactic radiosurgery; TPS = treatment planning system; TG = task group; VMAT = volumetric arc therapy.

* The initial version of this software was used with our treatment management system for decades. A new wrapper was added when the TPS was changed.

Table 4 Documentation recommendations and workflow to support safe implementation of in-house developed software

Documentation	Description	Considerations for environments with different resources
Policy on development and release for in-house developed software for clinical use	The policy supports transparency in the process. It should specify independent review of documentation before software is implemented in the clinic.	The requirement for a written policy demonstrates a commitment to patient safety and continuous quality improvement.
Design specifications and release notes	The software proposal typically specifies inputs, outputs, and intended users.	Discuss with primary intended users.
Release notes	These are created by the software engineer or lead developer.	If another physicist is unavailable to provide an independent review, a physician and dosimetrist could review the notes. Alternatively, the developer could reach out to a colleague at another institution for peer review.
Test suite list	Based on the functionality, a standard test suite should be created to test the limits of the software (see AAPM TG 53[14]).	The test suite can be created working with dosimetrists or other potential users.
Software commissioning report	A commissioning report should be created.	If an independent physicist is unavailable, then a dosimetrist or other personnel could perform testing and provide feedback. Regardless of the potential advantages for clinical use, software should not be released to the clinic until it has been properly commissioned and documented.
Training materials and/or documentation of training sessions	Training materials should be created for users. Users should be aware of the release version and the range of situations approved for clinical use.	If there are any exclusions or unsafe conditions, users should be made aware of those situations. It may be necessary to set up an audit program.
Event reporting	Event reporting supports a culture of safety and can be used to capture events related to the software, such as incorrect usage, ambiguity in communication, or software bugs.	All members of a department should be encouraged to support a culture of safety (see ASTRO's *Safety is No Accident*[29]).

Abbreviations: AAPM TG = American Association of Physicists in Medicine Task Group; ASTRO = American Society for Radiation Oncology.

Conclusions

Given the limited information available within our field to support the quality control aspects of developing robust software for the clinic, we believe that this work outlines a strong process to support clarity in the software release cycle for patient safety. The process incorporates teamwork, defined roles, documentation, and commissioning, as well as incorporation of incident learning in support of patient safety when any software is used for clinical care. In our work, we have followed the essential elements of a comprehensive quality assurance program outlined by AAPM Task Group 40,[13] software testing defined in AAPM Task Group 53,[14] and the principles of AAPM Task Group 100 regarding incorporating formal risk assessment.[30] Over the past 15 years, this process has supported software that has been used for all patients treated in our clinic.

Acknowledgments

The software release process was originally developed in support of NIH P01 CA059827-20. The authors gratefully acknowledge Benedick A. Fraass, Daniel L. McShan, Wayne Keranen, Peter L. Roberson, Randall K. Ten Haken, Paul G. Archer, Karen Vineberg, and the late Cheryl White for their contributions in developing this process when we had the UMPlan treatment planning system. The authors also gratefully acknowledge Eduardo Acosta and Carlos Anderson for their contributions to aspects of this work. We acknowledge department-wide support from employees and our clinical operations team in supporting these contributions for high quality patient care.

Supplementary materials

Supplementary material associated with this article can be found in the online version at doi:10.1016/j.adro.2021.100768.

References

1. McShan DL, Silverman A, Lanza DM, Reinstein LE, Glicksman AS. A computerized three-dimensional treatment planning system utilizing interactive colour graphics. *Br J Radiol*. 1979;52:478–481.
2. Mohan R, Barest G, Brewster LJ, et al. A comprehensive three-dimensional radiation treatment planning system. *Int J Radiat Oncol Biol Phys*. 1988;15:481–495.
3. Perez CA, Purdy JA, Harms W, et al. Three-dimensional treatment planning and conformal radiation therapy: Preliminary evaluation. *Radiother Oncol*. 1995;36:32–43.
4. Ten Haken RK, Fraass BA, Kessler ML, McShan DL. Aspects of enhanced three-dimensional radiotherapy treatment planning. *Bull Cancer*. 1995;82(Suppl 5):592s–600s.
5. Tewell MA, Adams R. The PLUNC 3D treatment planning system: A dynamic alternative to commercially available systems. *Med Dosim*. 2004;29:134–138.
6. Moore KL, Brame RS, Low DA, Mutic S. Experience-based quality control of clinical intensity-modulated radiotherapy planning. *Int J Radiat Oncol Biol Phys*. 2011;81:545–551.
7. Yang D, Wu Y, Brame RS, et al. Technical note: Electronic chart checks in a paperless radiation therapy clinic. *Med Phys*. 2012;39:4726–4732.
8. Olsen LA, Robinson CG, He GR, et al. Automated radiation therapy treatment plan workflow using a commercial application programming interface. *Pract Radiat Oncol*. 2014;4:358–367.
9. Mayo CS, Yao J, Eisbruch A, et al. Incorporating big data into treatment plan evaluation: Development of statistical DVH metrics and visualization dashboards. *Adv Radiat Oncol*. 2017;2:503–514.
10. Covington EL, Chen X, Younge KC, et al. Improving treatment plan evaluation with automation. *J Appl Clin Med Phys*. 2016;17:16–31.
11. Halabi T, Lu HM. Automating checks of plan check automation. *J Appl Clin Med Phys*. 2014;15:4889.
12. Hadley SW, Kessler ML, Litzenberg DW, et al. Streamlining and Automating QA in radiotherapy. *J Appl Clin Med Phys*. 2016;17:387–395.
13. Kutcher GJ, Coia L, Gillin M, et al. Comprehensive QA for radiation oncology: Report of AAPM Radiation Therapy Committee Task Group 40. *Med Phys*. 1994;21:581–618.
14. Fraass B, Doppke K, Hunt M, et al. American Association of Physicists in Medicine Radiation Therapy Committee Task Group 53: Quality assurance for clinical radiotherapy treatment planning. *Med Phys*. 1998;25:1773–1829.
15. FDA. Premarket notification 510(k). Available at: https://www.fda.gov/medical-devices/premarket-submissions/premarket-notification-510k. Accessed November 13, 2020.
16. *Eclipse Scripting API Reference Guide*. Finland: Varian Medical Systems; 2011.
17. *RayStation 4.5 Scripting Guideline*. Sweden: RaySearch Laboratories AB; 2014.
18. Paradis KC, Naheedy KW, Matuszak MM, Kashani R, Burger P, Moran JM. The fusion of incident learning and failure mode and effects analysis for data-driven patient safety improvements. *Pract Radiat Oncol*. 2021;11:e106–e113.
19. Fraass BA, McShan DL, Kessler ML. Computer-Controlled Treatment Delivery. *Semin Radiat Oncol*. 1995;5:77–85.
20. Fraass BA, McShan DL, Kessler ML, Matrone GM, Lewis JD, Weaver TA. A computer-controlled conformal radiotherapy system. *I: Overview Int J Radiat Oncol Biol Phys*. 1995;33:1139–1157.
21. Evans SB, Fraass BA, Berner P, et al. Standardizing dose prescriptions: An ASTRO white paper. *Pract Radiat Oncol*. 2016;6:e369–e381.
22. Bourne KC. Change control management. *Application Administrators Handbook: Installing, Updating, and Troubleshooting Software*. Morgan Kaufman; 2014:96–111.
23. Dooley J. *Software Development, Design and Coding*. 2nd ed. Apress; 2017. ISBN 978-1-4842-3152-4.
24. Paul M. *Seven Qualities of Highly Secure Software*. Boca Raton, FL: CRC Press; 2012.
25. Younge KC, Lee C, Moran JM, Feng M, Novelli P, Prisciandaro JI. Failure mode and effects analysis in a dual-product microsphere brachytherapy environment. *Pract Radiat Oncol*. 2016;6:e299–e306.
26. Hoopes DJ, Dicker AP, Eads NL, et al. RO-ILS: Radiation Oncology Incident Learning System: A report from the first year of experience. *Pract Radiat Oncol*. 2015;5:312–318.
27. Purdie TG, Dinniwell RE, Letourneau D, Hill C, Sharpe MB. Automated planning of tangential breast intensity-modulated radiotherapy using heuristic optimization. *Int J Radiat Oncol Biol Phys*. 2011;81:575–583.

28. Marks LB, Jackson M, Xie L, et al. The challenge of maximizing safety in radiation oncology. *Pract Radiat Oncol*. 2011;1:2–14.
29. American Society for Radiation Oncology. *Safety is No Accident: A Framework for Quality Radiation Oncology Care*. Arlington, VA: ASTRO; 2019. Available at: https://www.astro.org/Patient-Care-and-Research/Patient-Safety/Safety-is-no-Accident. Accessed September 3, 2021.
30. Huq MS, Fraass BA, Dunscombe PB, et al. The report of Task Group 100 of the AAPM: Application of risk analysis methods to radiation therapy quality management. *Med Phys*. 2016;43:4209.
31. Covington EL, Popple RA, Cardan RA. Technical note: Use of automation to eliminate shift errors. *J Appl Clin Med Phys*. 2020;21:192–195.
32. Ezzell G, Chera B, Dicker A, et al. Common error pathways seen in the RO-ILS data that demonstrate opportunities for improving treatment safety. *Pract Radiat Oncol*. 2018;8:123–132.
33. Siochi RA, Balter P, Bloch CD, et al. A rapid communication from the AAPM Task Group 201: Recommendations for the QA of external beam radiotherapy data transfer. AAPM TG 201: Quality assurance of external beam radiotherapy data transfer. *J Appl Clin Med Phys*. 2010;12:3479.
34. Zhang B, Chen S, Nichols E, D'Souza W, Prado K, Yi B. A practical cyberattack contingency plan for radiation oncology. *J Appl Clin Med Phys*. 2020;21:181–186.
35. Paradis KC, Mayo C, Owen D, et al. The special medical physics consult process for reirradiation patients. *Adv Radiat Oncol*. 2019;4:559–565.
36. Younge KC, Roberts D, Janes LA, Anderson C, Moran JM, Matuszak MM. Predicting deliverability of volumetric-modulated arc therapy (VMAT) plans using aperture complexity analysis. *J Appl Clin Med Phys*. 2016;17:124–131.
37. Simiele SJ, Chen X, Lee C, et al. Development and comprehensive commissioning of an automated brachytherapy plan checker. *Brachytherapy*. 2020;19:355–361.